Ethernet Protocol Stack Chip CH394

Version: 1.2

https://wch-ic.com

1. Overview

CH394 is an Ethernet protocol stack management chip, which is used for Ethernet communication in MCU system.

The CH394 chip comes with 10/100M Ethernet media transport layer MAC and physical layer transceiver PHY, fully compatible with IEEE802.3 protocols, and built-in Ethernet protocol stack firmware for IP, ARP, ICMP, IGMP, UDP, and TCP. MCU system can conveniently communicate with the network through CH394 chip. CH394 supports Wake-on-LAN (WOL) and power-down mode.

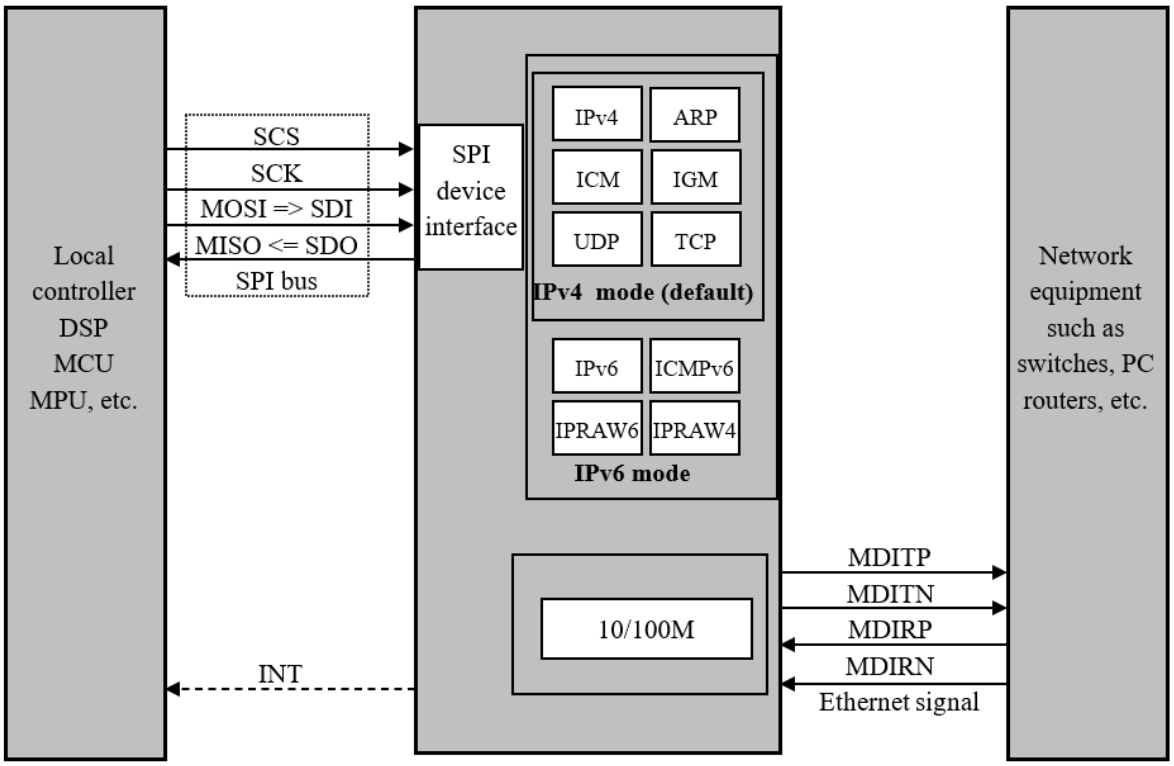
CH394Q has 2 modes: IPv4 mode and IPv6 mode. The default is IPv4 mode, which is compatible with various network protocols such as IPv4, ARP, ICMP, IGMP, UDP, TCP, etc. The IPv6 mode additionally supports protocols such as IPv6, ICMPv6, IPRAW4 and IPRAW6.

CH394Q provides SPI interface, CH394L provides SPI interface and 8-bit slave parallel interface. Controllers such as MCU/DSP/MPU can control CH394Q chip to communicate with Ethernet through SPI interface. Or through SPI interface or 8-bit parallel port to control CH394L chip for Ethernet communication.

The contents of this manual are mainly applicable to CH394Q and CH394L. For details, please refer to the CH394DS2 manual.

The following figure shows the application block diagram of CH394Q.

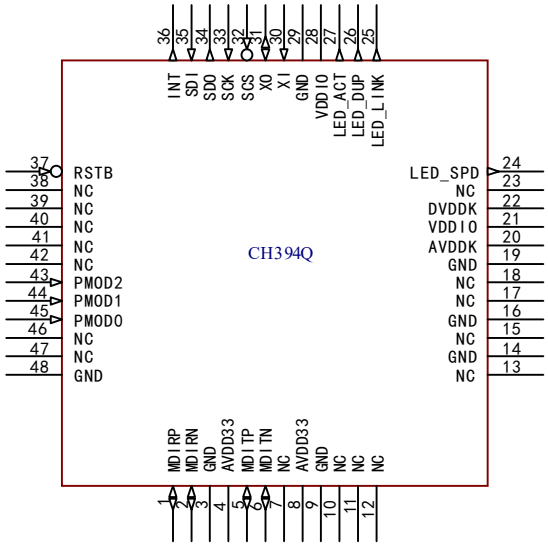
Figure 1-1 CH394Q application block diagram



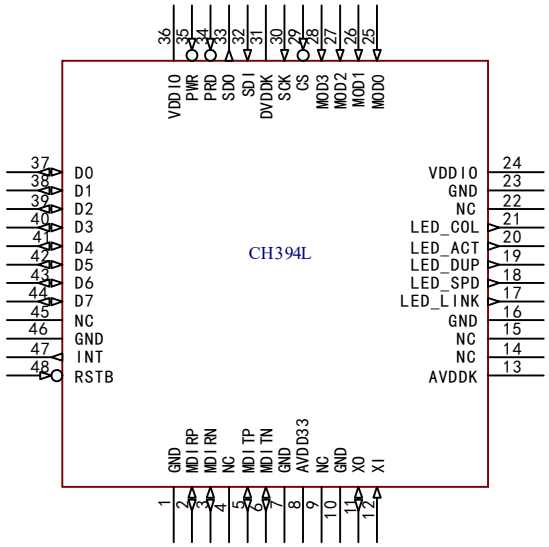
2. Features

- Built-in Ethernet Media Access Control (MAC) and Physical layer transceiver (PHY)
- Support 10/100M, full-duplex/half-duplex adaptation, and is compatible with IEEE802.3 protocol
- Support automatic conversion of MDI/MDIX lines
- I/O ports support 3.3V, 2.5V, 1.8V, 1.2V power supply, compatible with a variety of voltage standard microcontroller
- Built-in 50Ω impedance matching resistor, built-in 25MHz crystal oscillator capacitance, peripheral circuitry is streamlined.
- Built-in TCP/IP protocol suite. Support IPv4, ARP, ICMP, IGMP, UDP and TCP protocols
- CH394Q supports both IPv4 mode and IPv6 mode, and IPv4 mode is used by default.
- CH394Q supports 8 sockets and CH394L supports 4 sockets. Can transmit and receive data at the same time
- Support MACRAW mode and IPRAW mode
- Provide a SPI slave interface (SPI mode 0 or 3) up to 62MHz, with the high bit coming first
- The CH394L provides a high-speed 8-bit slave parallel interface that supports a parallel data bus connected to the microcontroller.
- Support network Wake-on-LAN (WOL) and power-down mode
- Provide LED status display (Link, ACT, 10/100M, full-duplex/half-duplex, etc.)
- Built-in 32KRAM can be used for Ethernet data transceiver, and each Socket transceiver buffer can be configured flexibly
- LQFP48 lead-free package

3. Pinouts



CH394Q



CH394L

Table 3-1 Package description

Package form	Body size	Pin pitch		Package description	Order model
LQFP48	7*7mm	0.5mm	19.7mil	Low Profile Quad Flat Pack	CH394Q
LQFP48	7*7mm	0.5mm	19.7mil	Low Profile Quad Flat Pack	CH394L

Note: See this datasheet for CH394Q pin definitions and Datasheet 2 for CH394L pin definitions.

4. CH394Q Pins Definition

Table 4-1 CH394Q pin definition

CH394Q Pin No.	Pin name	Type	Pin description
1	MDIRP	I/O	Differential input in 10BASE-T/100BASE-TX MDI mode; Differential output in 10BASE-T/100BASE-TX MDIX mode.
2	MDIRN		
5	MDITP	I/O	Differential output in 10BASE-T/100BASE-TX MDI mode; Differential input in 10BASE-T/100BASE-TX MDIX mode.
6	MDITN		
4	AVDD33	P	For 3.3V main power input, it is recommended to place 0.1uF in parallel with 10uF or 4.7uF ground capacitance close to the chip, or a single 1uF ~ 4.7uF.
8	AVDD33	P	3.3V power input, it is recommended to connect 0.1uF or 1uF capacitor to ground.
20	AVDDK	P	The external 1uF capacitor to ground is placed close to the chip.
21/28	VDDIO	P	For the power input of I/O interface, it is recommended to place 0.1uF or 1uF capacitance to ground close to the chip.
22	DVDDK	P	The external 0.1uF or 1uF capacitor to ground is placed close to the chip.
3/9/14/16/19/29/48	GND	P	Analog ground.
7/10/11/12/13/15/17/18/23/38/39/40/41/42/46/47	NC	-	Reserved, it is recommended to float.
24	LED_SPD	O	Network speed indicator LED output: Low level indicates 100Mbps; high level indicates 10Mbps.
25	LED_LINK	O	Network connection indication LED output: Low level indicates connected; high level indicates not connected.
26	LED_DUP	O	Duplex indicator LED output: Low level indicates full-duplex; high level indicates half-duplex.
27	LED_ACT	O	Carrier induction indicator LED output: A blinking LED indicates a carrier sense signal.
30	XI	I	Crystal oscillator input with external 25MHz crystal one end, or external clock input with built-in crystal matching capacitor.
31	XO	O	Crystal oscillator inverted output, need to be externally connected to the other end of the 25MHz crystal, built-in crystal matching capacitor.
32	SCS	I/PU/FT	SPI chip select input, active low.
33	SCK	I/FT	SPI clock input, supporting mode 0 or 3.

<table>
<tr><td>34</td><td>SDO</td><td>O/FT</td><td>SPI serial data output, connected to MISO of the host SPI interface, SDO is in high resistance state when SCS is high.</td></tr>
<tr><td>35</td><td>SDI</td><td>I/FT</td><td>SPI serial data input to the MOSI of the host SPI interface.</td></tr>
<tr><td>36</td><td>INT</td><td>O/FT</td><td>Interrupt request output, active low.</td></tr>
<tr><td>37</td><td>RSTB</td><td>I/PU/FT</td><td>Reset input, active low.</td></tr>
<tr><td>43</td><td>PMOD2</td><td>I/PU/FT</td><td rowspan="3">PHY operation mode selection:
The PMOD pin has a built-in pull-up resistor, when the PHY is in self-negotiation mode. If there is no special need, it is recommended to keep the PMOD pin dangling.
<table><tr><th>PMOD[2:0]</th><th>Description</th></tr><tr><td>000</td><td>10M half-duplex, auto-negotiation disabled.</td></tr><tr><td>001</td><td>10M full-duplex, auto-negotiation disabled.</td></tr><tr><td>010</td><td>100M half-duplex, auto-negotiation disabled.</td></tr><tr><td>011</td><td>100M full-duplex, auto-negotiation closed.</td></tr><tr><td>100</td><td>100M half-duplex, auto-negotiation enabled.</td></tr><tr><td>101</td><td>Reserved</td></tr><tr><td>110</td><td>Reserved</td></tr><tr><td>111</td><td>Start auto-negotiation (Recommended default mode)</td></tr></table></td></tr>
<tr><td>44</td><td>PMOD1</td><td>I/PU/FT</td></tr>
<tr><td>45</td><td>PMOD0</td><td>I/PU/FT</td></tr>
</table>

Note 1: I = Input; O = Output; I/O = Input/Output; FT = Tolerant 5V;
P = Power supply; PU = Built-in pull-up resistor.

5. CH394Q Data Format

The data in this manual have a suffix of B or b for binary numbers and H for hexadecimal numbers, otherwise they are decimal numbers. The CH394Q registers use big-end mode for data storage. In memory, the low byte is placed in the high address bit while the high byte is located in the low address bit.

5.1 Operation Mode

CH394Q can share the SPI interface with other SPI devices. The peripheral host only needs to pull down the SCS pin to communicate with CH394Q. The interface can be used by other devices during non-communication period.

5.2 Data Frame

5.2.1 Data Frame Format

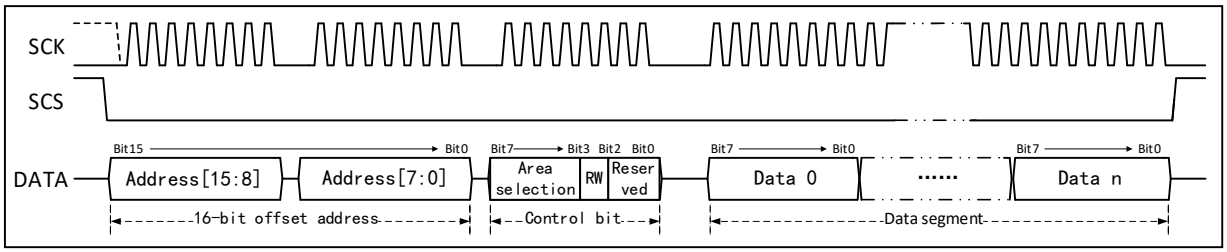
The CH394Q data frame consists of three parts: A 16-bit offset address, an 8-bit control word, and an N-byte data segment.

The offset address is the CH394Q register address or the offset address of the RX/TX buffer.

The control word is used to define the ownership of the offset area set in the address segment and determine the read/write mode.

Pulling SCS low indicates the start of a new data frame, and pulling SCS high indicates the end of a data frame.

Figure 5-1 CH394Q data frame



CH394Q supports continuous reading or writing of data, starting from the starting address. After each transmission of data at an offset address, the offset address will automatically increase by 1 to transmit the next data.

5.2.2 Control Phase

The CH394Q control phase specifies:

1. Ownership of the offset area set in the address segment;
2. SPI read/write mode;

Table 5-1 Control word

Bit	Name	Description
[7:3]	BS	Block selection bits: Used to define the ownership of the offset area set in the address segment. CH394Q contains 1 common register, 8 socket registers, and the corresponding socket read and write buffer area.

BS [7:3]	Description

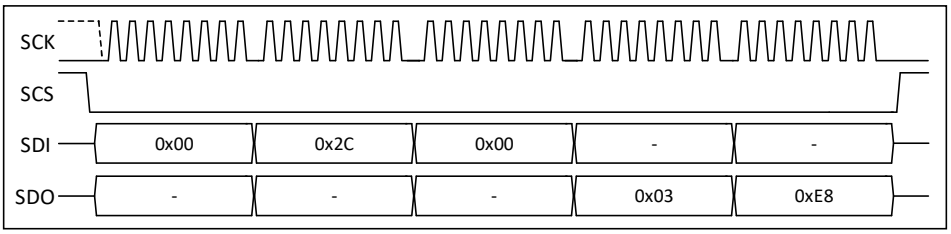
		Segment number selection[7:5] / Function[4:3] 000 00 Common register 000 01 Socket 0 register 000 10 Socket 0 TX buffer 000 11 Socket 0 RX buffer 001 00 Meaningless 001 01 Socket 1 register 001 10 Socket 1 TX buffer 001 11 Socket 1 RX buffer 010 00 Meaningless 010 01 Socket 2 register 010 10 Socket 2 TX buffer 010 11 Socket 2 RX buffer 011 00 Meaningless 011 01 Socket 3 register 011 10 Socket 3 TX buffer 011 11 Socket 3 RX buffer 100 00 Meaningless 100 01 Socket 4 register 100 10 Socket 4 TX buffer 100 11 Socket 4 RX buffer 101 00 Meaningless 101 01 Socket 5 register 101 10 Socket 5 TX buffer 101 11 Socket 5 RX buffer 110 00 Meaningless 110 01 Socket 6 register 110 10 Socket 6 TX buffer 110 11 Socket 6 RX buffer 111 00 Meaningless 111 01 Socket 7 register 111 10 Socket 7 TX buffer 111 11 Socket 7 RX buffer
2	RW	Read/write selection bit: 0: SPI read; 1: SPI write.
[1:0]	Reserved	Reserved, default is 0.

5.3 Data Frame Example

(1) Read 2-byte data

Read 2-byte of unreachable port information from the common register (Assuming the unreachable port is 0x03e8):

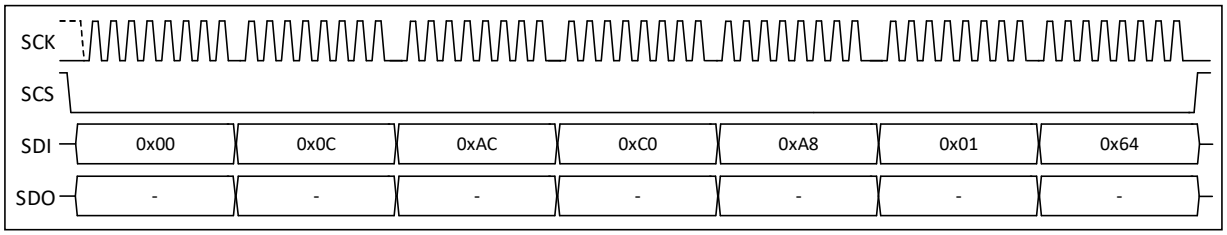
Figure 5-2 Example 1



(2) Write 4-byte data

Configure the Socket5 destination IP as 192.168.1.100:

Figure 5-3 Example 2



6. CH394Q Register and Buffer

CH394Q supports IPv4 mode and IPv6 mode. IPv4 mode is used by default, this chapter only introduces the IPv4 related functions of CH394Q, and all the protocols mentioned refer to IPv4. For more information about IPv6, please refer to Chapter 8.

CH394Q has 1 common register area, 8 socket register areas, 8 receive buffer areas and 8 transmit buffer areas. There are multiple configurable registers in each register area. Each register in the same area has a different offset address. The register to be read or written is determined by the different offset addresses in the address segment. The BS bit of the control word is used to define which register area the offset address set in the address segment belongs to.

CH394Q has a total of 16K transmit buffer, 8 sockets each default 2K; 16K receive buffer, 8 sockets each default 2K.

When reading/writing the socket receive/transmit buffer, the offset address in the address segment (Obtained by querying the RD/WR register) represents the starting address of the data storage, and the BS bit in the control word controls the specific receive/transmit buffer to be read/written.

6.1 Register Table

6.1.1 Common Register Table

The CH394Q common register configures the CH394Q mode, IP, MAC, etc. To read/write the common register, you need to fill in the offset address corresponding to the register in the offset address part of the data frame, and set the BS part of the control word to 00000b.

Table 6-1 Common register table

Offset address	Name	Access	Description	Reset value
0x0000	MODE	RW	Mode register	0x00
0x0001 … 0x0004	GWIP0 … GWIP3	RW	Gateway address register	0x00 … 0x00
0x0005 … 0x0008	SMIP0 … SMIP3	RW	Subnet mask register	0x00 … 0x00
0x0009 … 0x000E	MAC0 … MAC5	RW	MAC address register	0xXX … 0xXX
0x000F … 0x0012	IP0 … IP3	RW	Source IP address register	0x00 … 0x00
0x0013 0x0014	IIT0 IIT1	RW	Interrupt interval time register	0x00 0x00
0x0015	GINT	RW	Global interrupt register	0x00
0x0016	GINTE	RW	Global interrupt enable register	0x00
0x0017	SINT	RO	Socket interrupt register	0x00
0x0018	SINTE	RW	Socket interrupt enable register	0x00
0x0019	RTIME0	RW	Retransmission time register	0x07

0x001A	RTIME1			0xD0
0x001B	RCNT	RW	Retransmission count register	0x08
0x001C … 0x0027	-	-	Reserved	-
0x0028 … 0x002B	UNIP0 … UNIP3	RO	Unreachable IP register	0x00 … 0x00
0x002C 0x002D	UNPORT0 UNPORT1	RO	Unreachable port register	0x00 0x00
0x002E	PHY_CFG	RW	PHY configuration register	0xB8
0x002F … 0x0038	-	-	Reserved	-
0x0039	CHIPV	RO	Chip version register	0xXX
0x003E	CHIP_CONFIG	RW	Chip extended function register	0x00

6.1.2 Socket Register Table

CH394Q provides 8 Sockets, each Socket corresponds to a Socket register area, the register area corresponding to Socket n ($0 \leq n \leq 7$) is selected by the BS bit of the control word (Refer to 5.2.2 for details), and the register of Socket n to be read/written is determined by the offset address.

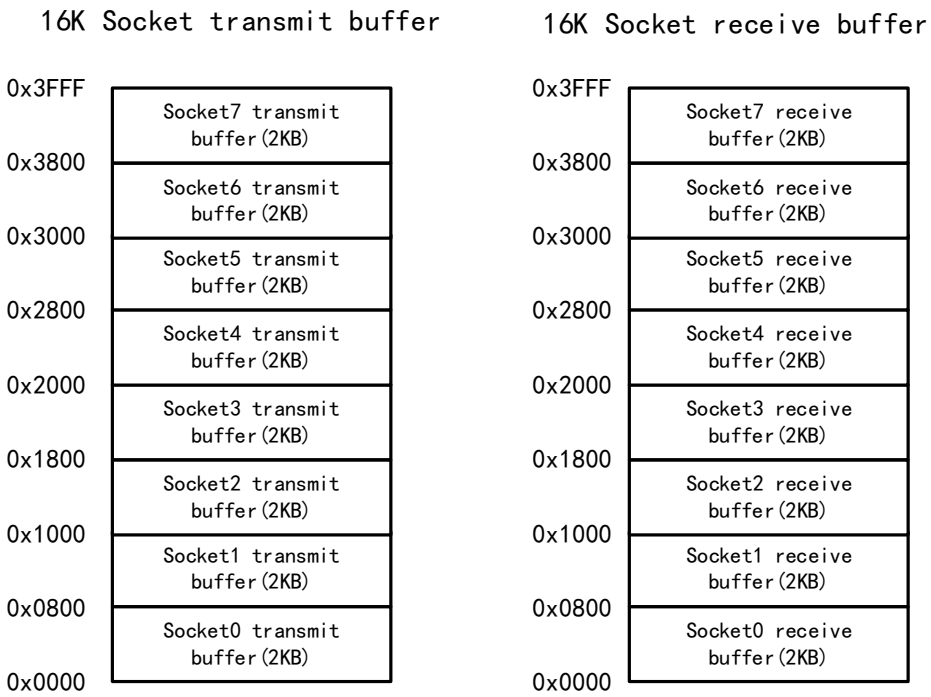
Table 6-2 Socket register table

Offset address	Name	Access	Description	Reset value
0x0000	Sn_MODE	RW	Socket n mode register	0x00
0x0001	Sn_CTRL	RW	Socket n control register	0x00
0x0002	Sn_INT	RW	Socket n interrupt register	0x00
0x0003	Sn_STA	RO	Socket n status register	0x00
0x0004 0x0005	Sn_PORT0 Sn_PORT1	RW	Socket n source port register	0x00 0x00
0x0006 … 0x000B	Sn_DMAC0 … Sn_DMAC5	RW	Socket n destination MAC register	0x00 … 0x00
0x000C … 0x000F	Sn_DIP0 … Sn_DIP3	RW	Socket n destination IP register	0x00 … 0x00
0x0010 0x0011	Sn_DPORT0 Sn_DPORT1	RW	Socket n destination port register	0x00 0x00
0x0012 0x0013	Sn_MTU0 Sn_MTU1	RW	Socket n maximum transfer unit register	0x00 0x00
0x0014	-	-	Reserved	-
0x0015	Sn_TOS	RW	Socket n IP service type register	0x00
0x0016	Sn_TTL	RW	Socket IP time-to-live register	0x80
0x0017	-	-	Reserved	-

… 0x001D				
0x001E	Sn_RXBUF_SIZE	RW	Socket n receive buffer size register	0x02
0x001F	Sn_TXBUF_SIZE	RW	Socket n transmit buffer size register	0x02
0x0020 0x0021	Sn_TX_FS0 Sn_TX_FS1	RO	Socket n idle transmit buffer length register	0x08 0x00
0x0022 0x0023	Sn_TX_RD0 Sn_TX_RD1	RO	Socket n transmit buffer read pointer register	0xXX
0x0024 0x0025	Sn_TX_WR0 Sn_TX_WR1	RW	Socket n transmit buffer write pointer register	0xXX
0x0026 0x0027	Sn_RX_RS0 Sn_RX_RS1	RO	Socket n receive data length register	0x00 0x00
0x0028 0x0029	Sn_RX_RD0 Sn_RX_RD1	RW	Socket n receive buffer read pointer register	0xXX
0x002A 0x002B	Sn_RX_WR0 Sn_RX_WR1	RO	Socket n receive buffer write pointer register	0xXX
0x002C	Sn_INTE	RW	Socket n interrupt enable register	0xFF
0x002D 0x002E	Sn_IPF0 Sn_IPF1	RW	Socket n IP fragment register	0x40 0x00
0x002F	Sn_KEEPALIVE	RW	Socket n KeepAlive time register	0x00

6.2 RX/TX Buffer

Figure 6-1 RX/TX buffer



The CH394Q provides independent transmit and receive memory buffers for each socket n (n = 0 to 7), with a total of 16K receive buffers and 16K transmit buffers.

In the initial state, the transmitting and receiving buffers of each Socket are allocated to 2KB (The receiving

buffer is 16KB in total and the transmitting buffer is 16KB in total). Users can reallocate the 16KB memory resources to each Socket by configuring the Socket transmit buffer size register (Sn_TXBUF_SIZE) and the Socket receive buffer size register (Sn_RXBUF_SIZE) as required, but it is necessary to ensure that the sum of the receive buffer and transmit buffer sizes of all sockets does not exceed 16KB to prevent data transmission errors.

7. CH394Q Register Description

This chapter only describes the IPv4-related functions of the CH394Q, and the protocols mentioned refer to IPv4. For more information about IPv6, please refer to Chapter 8.

Socket Pair is a quaternion containing the Source IP, Source Port, Destination IP, and Destination Port that uniquely identifies both sides of a connection on the Internet. This manual is referred to as Socket for short. CH394Q can provide 8 Socket at the same time, and their index values are 0, 1, 2, 3 ....

The high and low bytes of IP and MAC addresses agreed in this manual may be different from some documents, just for convenience:

For example, the IP address is 192.168.1.2, where 192 is the highest byte and 2 is the lowest byte.

For example, the MAC address is 0x11:0x22:0x33:0x44:0x55:0x66, where 0x11 is the highest byte and 0x66 is the lowest byte.

Abbreviated description of the bit attribute in the register:

Register bit attributes	Property description
RF	Read-only attribute, read a fixed value.
RO	Read-only attribute, changed by hardware.
RZ	Read-only attribute, auto bit clear 0 after read operation.
WO	Write only attribute (not readable, read value uncertain)
WA	Write-only attribute, writable in Safe mode.
WZ	Write only attribute, auto bit clear 0 after write operation.
RW	Readable and writable.
RWA	Readable, writable in Safe mode.
RW1	Readable, write 1 valid, write 0 invalid.
RW0	Readable, write 0 valid, write 1 invalid.
RW1T	Readable, write 0 invalid, write 1 flipped.
RW1Z	Readable, write 1 clear, write 0 invalid.

7.1 Common Register Description

CH394Q register uses big terminal mode for data storage. In memory, the low byte is placed in the high address bit, while the high byte is placed in the low address bit.

7.1.1 Mode Register (MODE) [0x0000]

This register is used to control the operation mode.

Table 7-1 Mode Register

Bit	Name	Description	Access	Default value
7	RST	Set 1 for software reset, and it will be cleared automatically after the reset is completed.	RW	0
6	Reserved	Reserved	RO	0
5	WOL	Wake on LAN enabled: 0: Disable WOL mode;	RW	0

		1. Enable WOL mode. CH394Q integrates the magic packet WOL mode. After enabling the magic packet wake-up, CH394Q will generate WOL interrupt after receiving the magic packet (0xFFFFFFFFFFFF+16 CH394Q MAC addresses) that matches its own MAC address from UDP.		
4	PB	PING packet block enabled: 0: Disable PING block; 1: Enable PING block.	RW	0
3	Reserved	Reserved	RO	0
2	Reserved	Reserved	RO	0
1	FARP	Forced ARP enable: 0: Disable forced ARP mode; 1: Enable forced ARP mode. Turn on forced ARP mode. When forced ARP is turned on, an ARP request will be forced to be sent before each UDP packet is sent.	RW	0
0	Reserved	Reserved	RO	0

7.1.2 Gateway IP Address Register (GWIP) [0x0001-0x0004]

This register is used to set the gateway address, and the low byte address stores the high byte of the gateway address.

For example: The gateway IP is 192.168.1.1.

Address	0x0001	0x0002	0x0003	0x0004
Data	0xC0	0xA8	0x01	0x01

7.1.3 Subnet Mask IP Address Register (SMIP) [0x0005-0x0008]

This register is used to set the subnet mask address, and the low byte address stores the high byte of the gateway address.

For example: Subnet mask IP 255.255.255.0.

Address	0x0005	0x0006	0x0007	0x0008
Data	0xFF	0xFF	0xFF	0x00

7.1.4 MAC Address Register (MAC) [0x0009-0x000E]

This register is used to set the MAC address of CH394Q, and the low byte address stores the high byte of the MAC address.

The CH394Q chip has burned the MAC address assigned by IEEE when it leaves the factory. Please do not set the MAC address unless it is necessary.

For example: The MAC address is 0x38:0x3B:0x26:0x11:0x22:0x33.

Address	0x0009	0x000A	0x000B	0x000C	0x000D	0x000E
Data	0x38	0x3B	0x26	0x11	0x22	0x33

7.1.5 Source IP Address Register (IP) [0x000F-0x0012]

This register is used to set the IP address of CH394Q, and the low byte address stores the high byte of the IP address.

For example: The source address is 192.168.1.100

Address	0x000F	0x0010	0x0011	0x0012
Data	0xC0	0xA8	0x01	0x64

7.1.6 Interrupt Interval Register (IIT) [0x0013-0x0014]

This register is used to set the waiting time for the interrupt to take effect, and the default value is 0. When all interrupt flags are disposed of, the INT signal line returns to a high state, and a set IIT interval needs to elapse before the CH394Q notifies the next interrupt and pulls the INT signal line low.

The formula for calculating the interval time is: Time = 1/120000000*4*(IIT+1).

For example, if the register value is set to 1000, then the interval time is about 33.3us

Address	0x0013	0x0014
Data	0x03	0xE8

7.1.7 Global Interrupt Register (GINT) [0x0015]

This register is used to obtain the global interrupt status. When an interrupt event occurs, the corresponding bit in the GINT register will be set to 1. Writing a 1 to this bit can clear the interrupt flag bit. If the enable bit of this interrupt in GINTE is also 1, the level of the INT pin becomes low. Thereafter, once the corresponding bit of the event is cleared or the corresponding interrupt enable bit is set to 0, and the SINT is 0, the INT pin is restored to high level.

Table 7-2 Global interrupt register

Bit	Name	Description	Access	Default value
7	IP_CONFLI	IP conflict interrupt: This interrupt occurs when CH394Q detects that its own IP address is the same as that of other network devices in the same network segment.	RW1Z	0
6	UNREACH	Unreachable interrupt: When CH394Q receives the ICMP unreachable interrupt message, it saves the IP address, port and protocol type of the unreachable IP packet in the unreachable information table, and then generates this interrupt. After receiving this interrupt, user can query UNIP and UNPORT registers to obtain unreachable information.	RW1Z	0
5	Reserved	Reserved	RO	0
4	MP	Magic packet wake-up interrupt: This interrupt occurs when the Magic Packet function is enabled and the Magic Packet WOL mode is	RW1Z	0

		received through UDP.		
[3:0]	Reserved	Reserved	RO	0

7.1.8 Global Interrupt Enable Register (GINTE) [0x0016]

This register is used to control which interrupt sources can trigger interrupts, and each interrupt event enable bit corresponds to one bit of the global interrupt register (GINT).

When the interrupt event is generated, only when the corresponding bit in GINTE is 1, CH394Q will pull down the INT pin and generate an interrupt.

Table 7-3 Global interrupt enable register

Bit	Name	Description	Access	Default value
7	IP_CONFLI	IP conflict interrupt enable bit: Enable this bit allows IP collision interrupts to be generated.	RW	0
6	UNREACH	Unreachable interrupt enable bit: Enable this bit allows unreachable interrupts to be generated.	RW	0
5	Reserved	Reserved.	RO	0
4	MP	Magic Packet Wake-up Interrupt Enable Bit: Enable this bit allows the generation of Magic Packet Wake-up Interrupt.	RW	0
[3:0]	Reserved	Reserved.	RO	0

7.1.9 Socket Interrupt Register (SINT) [0x0017]

This register is used to get the Socket interrupt status. When an interrupt event occurs in Socket n, the Sn_INT register corresponds to position 1 and the nth position of SINT is 1. The INT pin will also be pulled low when the corresponding bit of SINTE is 1. When Sn_INT is 0, the corresponding bit of SINT will also be cleared. The INT pin is pulled high when both SINT and GINT are '0x00'.

Table 7-4 Socket interrupt register

Bit	Name	Description	Access	Default value
7	S7_INT	Socket7 interrupt.	RO	0
6	S6_INT	Socket6 interrupt.	RO	0
5	S5_INT	Socket5 interrupt.	RO	0
4	S4_INT	Socket4 interrupt.	RO	0
3	S3_INT	Socket3 interrupt.	RO	0
2	S2_INT	Socket2 interrupt.	RO	0
1	S1_INT	Socket1 interrupt.	RO	0
0	S0_INT	Socket0 interrupt.	RO	0

7.1.10 Socket Interrupt Enable Register (SINTE) [0x0018]

This register is used to control which Sockets can trigger interrupts, and each enable bit corresponds to one bit of the socket interrupt register (SINT).

When the Socket interrupt event is generated, only when the corresponding bit in SINTE is 1, CH394Q will pull down the INT pin and generate an interrupt.

Table 7-5 Socket interrupt enable register

Bit	Name	Description	Access	Default value
7	S7_INT	Socket7 interrupt enable bit: Enabling this bit allows the generation of Socket 7 interrupts.	RW	0
6	S6_INT	Socket6 interrupt enable bit: Enabling this bit allows the generation of Socket 6 interrupts.	RW	0
5	S5_INT	Socket5 interrupt enable bit: Enabling this bit allows the generation of Socket 5 interrupts.	RW	0
4	S4_INT	Socket4 interrupt enable bit: Enabling this bit allows the generation of Socket 4 interrupts.	RW	0
3	S3_INT	Socket3 interrupt enable bit: Enabling this bit allows the generation of Socket 3 interrupts.	RW	0
2	S2_INT	Socket2 interrupt enable bit: Enabling this bit allows the generation of Socket 2 interrupts.	RW	0
1	S1_INT	Socket1 interrupt enable bit: Enabling this bit allows the generation of Socket 1 interrupts.	RW	0
0	S0_INT	Socket0 interrupt enable bit: Enabling this bit allows the generation of Socket 0 interrupts.	RW	0

7.1.11 Retransmission Time Register (RTIME) [0x0019-0x001A]

This register is used to set the retransmission time in TCP communication and the retransmission time of ARP. If TCP transmission or ARP request does not receive a response from the other party within the RTIME time, a retransmission operation will be performed or a timeout interrupt will be triggered.

RTIME requires two bytes of time value in 0.1ms. the default value is 2000 (0x07D0), i.e. 200ms (2000*0.1ms), RTIME register can be set to any value, CH394Q will automatically round down to a multiple of 1000.

For example: Set retransmission time as 5000 (0x1388), i.e. 500ms.

Address	0x0019	0x001A
Data	0x13	0x88

7.1.12 Retransmission Count Register (RCN) [0x001B]

This register is used to set the retransmission timeout count. The retransmission timeout count refers to the maximum number of times a packet is retransmitted without a response, and the default value is 8. When the retransmission exceeds the set number of times, the corresponding socket timeout interrupt bit will be set to 1.

7.1.13 Unreachable IP Register (UNIP) [0x0028-0x002B]

This register is used to obtain an unreachable IP address. When CH394Q transmits data to an unreachable port, the other party will reply with an ICMP packet (Destination port unreachable). After receiving the unreachable message, CH394Q will generate an unreachable interrupt. The microcontroller can use this register to obtain the unreachable address.

For example: Unreachable IP is 192.168.1.200

Address	0x0028	0x0029	0x002A	0x002B
Data	0xC0	0xA8	0x01	0xC8

7.1.14 Unreachable Port Register (UNPORT) [0x002C-0x002D]

This register is used to obtain an unreachable port. When CH394Q transmits data to an unreachable port, the other party will reply with an ICMP packet (Destination port unreachable). After receiving the unreachable message, CH394Q will generate an unreachable interrupt. The microcontroller can use this register to obtain the unreachable port.

For example: Unreachable Port is 2000

Address	0x002C	0x002D
Data	0x07	0xD0

7.1.15 PHY Configuration Register (PHY_CFG) [0x002E]

This register configures status of PHY, and it indicates the current PHY connection status.

Table 7-6 PHY configuration register

Bit	Name	Description	Access	Default value
7	RST	Reset: 0: PHY reset; 1: Normal work. *Note: This bit needs to be set to 1 manually after the PHY reset is completed.*	RW	1
6	OPWD	Operation mode selection bit: 0: Configure the PHY mode through hardware pins (PMOD[2:0]); 1: Configure the PHY mode via the PHY Configuration Register [5:3] bits. When CH394Q is powered on or reset by RSTB pin, it detects the level status of PMOD[2:0] pin to set the PHY operation mode. When the user wants to control the PHY mode through OPMDC, it is necessary to: 1. Set OPWD as 1; 2. Set OPMDC to the desired mode value. When the user wants to use the level state of the PMOD[2:0] pin to set the PHY operation mode again, he only needs to set the OPWD bit from 1 to 0 without powering on or resetting it again.	RW	0
[5:3]	OPMDC	PHY mode configuration bit	RW	111b

		[5:3] / Description 000: 10M half-duplex, auto-negotiation disabled 001: 10M full-duplex, auto-negotiation disabled 010: 100M half-duplex, auto-negotiation disabled 011: 100M full-duplex, auto-negotiation disabled 100: 100M half-duplex, auto-negotiation enabled 101: Reserved 110: Power-down mode 111: Enable auto-negotiation Defaults to self-negotiation mode.		
2	DUPS	Duplex status 0: half-duplex; 1: full-duplex.	RO	0
1	SPDS	Speed status 0: 10Mb/s; 1: 100Mb/s.	RO	0
0	LINKS	Link status 0: The physical layer does not establish a link; 1: The physical layer establishes the link.	RO	0

7.1.16 Chip Version Register (CHIPV) [0x0039]

This register is used to obtain the chip version.

7.1.17 Chip Extension Function (CHIP_CONFIG) [0x003E]

This register is used to control the configuration of the CH394Q's extended functions.

Table 7-7 Global interrupt register (IPv6)

Bit	Name	Description	Access	Default value
[7:1]	Reserved	Reserved	RO	0
0	IP_VER_SEL	IP protocol version selection 0: Use IPv4 mode; 1: Use IPv6 mode.	RW	0

7.2 Socket Register Description

7.2.1 Socket n Mode Register (Sn_MODE) [0x0000]

This register is used to configure the operation mode of Socket n.

Table 7-8 Socket n mode register

Bit	Name	Description	Access	Default value
7	MUL_MFEN	When Socket n works in UDP mode: 0: Disable multicast mode; 1: Enable multicast mode. When using multicast mode, you need to configure multicast IP and multicast MAC in advance through Sn_DIP and Sn_DMAC before the socket is opened. When Socket n is operating in MACRAW mode: 0: Disables MAC address filtering; 1: Enables MAC address filtering. When MACRAW mode is used, with MAC address filtering turned on, Socket n will only receive packets on the network whose destination MAC is the CH394Q's own MAC address as well as the broadcast address.	RW	0
6	BCASTB	Broadcast blocking shielding bit: 0: Disable broadcast blocking; 1: Enable broadcast blocking. After this position 1, in UDP and MACRAW mode, Socket n will not receive broadcast blocking on the network.	RW	0
5	NA_MV_MM	When Socket n is operating in TCP mode: 0: Disable no-delay ACK; 1: Enable no-delay ACK. After enabling no-delay ACK, Socket n will reply to ACK as soon as possible after receiving the packet from the opposite end without delay; otherwise, it will wait for RTIME before replying to ACK. When Socket n is operating in UDP multicast mode: 0: Use IGMP version 2; 1: Use IGMP version 1. When Socket n is operating in MACRAW mode: 0: Disable multicast blocking; 1: Enable multicast blocking. When this position is set to 1, in MACRAW mode, Socket n will not receive multicast packets on the network.	RW	0
4	UCASTB_MIP6B	When Socket n is operating in UDP multicast mode: 0: Disable unicast blocking; 1: Enable unicast blocking. When this position is set to 1, in UDP multicast mode, Socket n will not receive unicast packets on the network.	RW	0

<table>
<tr><td></td><td></td><td>When Socket n is working in MACRAW mode:
0: Disable IPv6 blocking;
1: Enable IPv6 blocking.
After this position 1, in MACRAW mode, Socket n will not receive IPv6 packets on the network.</td><td></td><td></td></tr>
<tr><td>[3:0]</td><td>PMD</td><td>Set Socket operation mode:
<table><tr><th>[3:0]</th><th>Description</th></tr><tr><td>0000</td><td>OFF</td></tr><tr><td>0001</td><td>TCP</td></tr><tr><td>0010</td><td>UDP</td></tr><tr><td>0100</td><td>MACRAW</td></tr></table>Note: Only Socket 0 can use MACRAW mode.</td><td>RW</td><td>0</td></tr>
</table>

7.2.2 Socket n Control Register (Sn_CTRL) [0x0001]

This register is used to set the control command of Socket n. After configuring this register, the register will be automatically cleared. After setting the control command, the control command usually takes a certain amount of time to execute. Users can judge the command execution status through the Sn_INT or Sn_STA register.

Table 7-9 Socket n mode register

<table>
<tr><th>Code</th><th>Name</th><th>Description</th></tr>
<tr><td>01H</td><td>OPEN</td><td>Open Socket n:
Initialize and open Socket n according to the mode set by Sn_MODE[3:0]. You can query the command execution status through Sn_STA, and when the command is executed successfully, the corresponding Sn_STA value should be:
<table><tr><th>Sn_MODE[3:0]</th><th>Sn_STA</th></tr><tr><td>0000 (OFF)</td><td>-</td></tr><tr><td>0001 (TCP)</td><td>0x13 (INIT)</td></tr><tr><td>0010 (UDP)</td><td>0x22 (UDP)</td></tr><tr><td>0100 (MACRAW)</td><td>0x42 (MACRAW)</td></tr></table></td></tr>
<tr><td>02H</td><td>LISTEN</td><td>Set Socket n to the listening state:
This command takes effect only when Socket n is in TCP mode and is in the INIT state.
When Socket n is set to LISTEN state, Socket n is in TCP Server and waits for the client to connect, and the Sn_STA register becomes 0x14 (LISTEN). After the client connects successfully, the Sn_STA register value becomes 0x17 (ESTABLISHED) and Sn_INT[0] is set to 1.
If the connection fails the Sn_STA register value becomes 0x00 (CLOSE) and Sn_INT[3] is set to 1.</td></tr>
<tr><td>04H</td><td>CONNECT</td><td>Enable Socket n to enter connection mode:
This command takes effect only when Socket n is in TCP mode and is in INIT state.
Enabling Socket to enter connection mode means TCP Client mode. When this command is executed, Socket n connects to the server according to the values</td></tr>
</table>

		set by Sn_DIP and Sn_DPORT, and the value of Sn_STA register changes to 0x17 (ESTABLISHED) and Sn_INT[0] is set to 1 when the connection is successful. When the socket connection fails: 1: ARP failure, cannot get the destination MAC address (Sn_INT[3] set to 1); 2: Connection timeout, cannot receive SYN/ACK packet from the other party after transmitting SYN packet, after retransmission timeout (Sn_INT[3] set to 1); 3: Receiving RST packet, being disconnected by the other party on its own initiative. The Sn_STA register value becomes 0x00 (CLOSE) when the above situation occurs.
08H	DISCONNECT	Socket n disconnect: This command is effective only when Socket n is in TCP mode. Active close: Actively transmit a FIN packet to the other end device; Passive close: After receiving a FIN packet from the other end, execute this command to reply a FIN packet to the other end. If the FIN packet does not receive the ACK response from the other side, after the retransmission timeout, Sn_INT[3] is set to 1, and the value of Sn_STA register becomes 0x00(CLOSE).
10H	CLOSE	Close Socket n: Directly closes the socket. When Socket n is in TCP mode, this command also directly closes the socket without transmitting a FIN packet.
20H	SEND	Socket n transmitting data: Transmitting the data in the Socket n transmitting buffer. See section 12.2.3 Data transmitting process for details.
21H	SNED_MAC	Socket n transmits data without ARP: This command only takes effect when Socket n is in UDP mode. The SEND process is the same as the transmit command, but the MAC address of the opposite end is not obtained through ARP before the contract is sent, and the MAC address set by Sn_DMAC is used directly.
22H	SEND_KEEP	Socket n transmits KeepAlive package: This command only takes effect when Socket n is in TCP mode. Actively transmit a heartbeat packet to the opposite device. If no reply is obtained after the timeout, Sn_INT[3] is set to 1, and the value of Sn_STA register becomes 0x00(CLOSE).
40H	RECV	Socket n receiving data: Complete the process of Socket n receiving data. For detailed steps, please refer to Chapter 12.2.2 Data Receiving Process.

7.2.3 Socket n Interrupt Register (Sn_INT) [0x0002]

This register is used to obtain the interrupt status of Socket n. When the Socket n interrupt event occurs, the corresponding bit of the Sn_INT register will be set to 1, and writing a 1 to this bit can clear the interrupt flag bit.

Table 7-10 Socket n interrupt register

Bit	Name	Description	Access	Default value
[7:5]	Reserved	Reserved.	RO	0
4	INT_SEND_SUC	Transmit completion interrupt: This interrupt is generated when the Socket n SEND command is completed.	RW1Z	0
3	INT_TIMEOUT	Timeout interrupt: This interrupt is generated when Socket n ARP times out or TCP retransmission times out.	RW1Z	0
2	INT_RECV	Receive interrupt: This interrupt is generated when Socket n receives a packet.	RW1Z	0
1	INT_DISCONNECT	Disconnect interrupt: This interrupt is generated when Socket n receives the other party's FIN or FIN/ACK or RST packet.	RW1Z	0
0	INT_CONNECT	Connection interrupt: This interrupt is generated when the Socket n is successfully connected with the opposite end.	RW1Z	0

7.2.4 Socket n Status Register (Sn_STA) [0x0003]

This register is used to obtain the current state of Socket n and the temporary state in operation.

Table 7-11 Socket n status values

Code	Name	Description
00H	CLOSE	Socket n is in the closed state.
13H	INIT	TCP Ready: When Socket n is in TCP mode and open is successful, Socket n is in INIT state. Only when Socket n is in INIT state, users can use LISTEN or CONNECT command to make the next connection.
14H	LISTEN	TCP Listening: This status indicates that Socket n is in listening state as a TCP server, waiting for TCP client connection. Sn_STA will become 0x17 (ESTABLISHED) after successful connection.
17H	ESTABLISHED	Socket establishes connection: When Socket is in the LISTEN state, it is connected successfully by the opposite TCP client as a TCP server mode; Or use the CONNECT command as a TCP client to connect successfully, and Sn_STA will become 0x17(ESTABLISHED). When Sn _ STA is 0x17 (ESTABLISHED), the connection is successful, and data transmission can be carried out normally at this time.
1CH	CLOSE_WAIT	Close Waiting: When a disconnect request packet (FIN packet) is received from the other party after a Socket n TCP connection is established, Sn_STA changes to

		0x10 (CLOSE_WAIT). At this time, the TCP connection is in a semi-disconnected state. If you want to completely disconnect, you need to use DISCONNT command, and if you want to CLOSE it directly, you can use Close command.
22H	UDP	Socket n is in UDP mode.
42H	MACRAW	Socket n is in MACRAW mode. *Note: Only Socket 0 can use MACRAW mode.*

Temporary state during TCP connection:

Table 7-12 Socket n TCP temporary status value

Code	Name	Description
15H	SYN_SENT	SYN request sent: When the CONNECT command is used, Socket n will transmit a packet of SYN request to the opposite end actively, and at this time, Sn_STA will become SYN_SENT. When the opposite end replies to the SYN/ACK packet, the connection is successfully established, and Sn_STA will become 0x17(ESTABLISHED). If there is no reply to the SYN/ACK packet at the opposite end, after the SYN request retransmission times out, Sn_INT[3] is set to 1, and the value of Sn_STA register becomes 0x00(CLOSE).
16H	SYN_RECV	SYN request received: When Socket n is in TCP Server mode in listening state, Sn_STA becomes SYN_RECV after receiving the connection request packet from the opposite end. At this time, Socket n will reply the SYN/ACK packet to the opposite end. After success, Sn_STA will become 0x17(ESTABLISHED). If the other party fails to reply to the ACK in time after transmitting the SYN/ACK packet, resulting in the retransmission timeout, then Sn_INT[3] is set to 1, and the value of Sn_STA register becomes 0x00(CLOSE).
18H	FIN_WAIT	These states all indicate that Socket n is closing. Generally, there is no need to care about these States in application, and the chip will automatically process and update. When the Socket n is closed successfully, or after the timeout (Sn_INT[3] is set to 1), the value of Sn_STA register becomes 0x00(CLOSE).
1BH	TIME_WAIT	
1DH	LAST_ACK	

7.2.5 Socket n Source Port Register (Sn_PORT) [0x0004-0x0005]

This register is used to set the source port number of Socket n.

For example: The source port number is 2000.

Address	0x0004	0x0005
Data	0x07	0xD0

7.2.6 Socket n Destination MAC Register (Sn_DMAC) [0x0006-0x000B]

This register is used to get or set the destination MAC address of Socket n. This register can be used in two ways:

1: Get the target MAC address. After ARP is successful, this register stores the MAC address obtained in the process of Socket n ARP.

2: Set the target MAC address. SNED_MAC command in UDP mode will be sent directly according to the set MAC address. When in UDP multicast mode, this register is used to configure the multicast MAC.

For example: The MAC address is 0x38:0x3B:0x26:0x44:0x55:0x66.

Address	0x0006	0x0007	0x0008	0x0009	0x000A	0x000B
Data	0x38	0x3B	0x26	0x44	0x55	0x66

7.2.7 Socket n Destination IP Register (Sn_DIP) [0x000C-0x000F]

This register is used to get or set the destination IP address of Socket n. This register can be used in two ways:

1. Get the target IP address. In TCP Server mode, after the connection is successful, this register stores the IP address of the TCP client.

2: Set the destination IP address, which is used to configure the destination IP address in UDP mode, TCP client mode. when UDP multicast mode, this register is used to configure the multicast IP.

For example: The destination IP address is 192.168.1.200.

Address	0x000C	0x000D	0x000E	0x000F
Data	0xC0	0xA8	0x01	0xC8

7.2.8 Socket n Destination Port Register (Sn_DPORT) [0x0010-0x0011]

This register is used to get or set the destination IP port of Socket n. There are 2 ways to use this register:

1. Get the destination port. In TCP Server mode, after the connection is successful, this register stores the port number of the TCP client.

2: Set the destination port, in TCP Client mode this value must be set, in UDP mode, the destination port number needs to be written to this register before transmitting.

For example: The target port number is 3000.

Address	0x0010	0x0011
Data	0x0B	0xB8

7.2.9 Socket n Maximum Transfer Unit Register (Sn_MTU) [0x0012-0x0013]

This register is used to set the maximum transmission unit (MTU) of Socket n, which takes effect in TCP mode and UDP mode. When the transmission length of user data exceeds the preset MTU size, the CH394Q built-in protocol stack will automatically perform data segmentation to ensure that each packet of data does not exceed the set value of MTU.

For example: Set MTU as 1000.

Address	0x0012	0x0013
Data	0x03	0xE8

7.2.10 Socket n IP Service Type Register (Sn_TOS) [0x0015]

This register is used to set the TOS service field in the IP header of the IP layer, which should be set before the Socket n OPEN. The default value is 0, and it does not need to be set by default.

7.2.11 Socket n IP Time-to-Life Register (Sn_TTL) [0x0016]

This register is used to set the TTL service field in the IP header of the IP layer and should be set before Socket n OPEN. The default value is 0x80, the default value is not required, and the maximum value is 255.

7.2.12 Socket n RX Buffer Size Register (Sn_RXBUF_SIZE) [0x001E]

This register is used to set the receive buffer size of Socket n, and the default value is 2(2KB). The register value can be set from 0 to 16, corresponding to 0 to 16KB. CH394Q allocates 2K space for each Socket by default.

Socket 0 receive buffer		Socket 1 receive buffer			Socket 7 receive buffer	
Block 0	Block 1	Block 2	Block 3	……	Block 14	Block 15
2K		2K			2K	

The default allocation of the internal receiving buffer of CH394Q is as shown in the figure above, which consists of 16 blocks in total, and the length of each block is 1024 bytes. MCU can freely allocate the size of each Socket receiving buffer. After the configuration of Sn_RXBUF_SIZE is completed, the receive buffer will be reallocated in the order of Socket0 0 to 7. When setting the receiving buffer, it should be noted that the sum of all Socket receiving buffers cannot exceed 16KB.

7.2.13 Socket n TX Buffer Size Register (Sn_TXBUF_SIZE) [0x001F]

This register is used to set the transmit buffer size of Socket n, and the default value is 2(2KB). The register value can be set from 0 to 16, corresponding to 0 to 16KB. CH394Q allocates 2K space for each Socket by default.

Socket 0 receive buffer		Socket 1 receive buffer			Socket 7 receive buffer	
Block 0	Block 1	Block 2	Block 3	……	Block 14	Block 15
2K		2K			2K	

The default allocation of CH394Q's internal transmit buffer is as shown above, which consists of 16 blocks, each of which is 1024 bytes long. MCU can freely allocate the size of each Socket transmitting buffer. After the configuration of Sn_TXBUF_SIZE is completed, the TX buffer will be reallocated in the order of Socket0 0 to 7. When setting the transmit buffer, it should be noted that the sum of all Socket transmit buffers cannot exceed 16KB.

7.2.14 Socket n TX Free Size Register (Sn_TX_FS) [0x0020-0x0021]

This register is used to obtain the size of the free space in the transmit buffer of Socket n. When no packet is sent, the free length is the size of the transmitted buffer of Socket n.

Before transmitting data, the user must query this register to obtain the size of the free space (See section 12.2.3 for details), and the length of the transmitted packet shall not be greater than the size of the free space.

For example: The length of the free buffer is 1000.

Address	0x0020	0x0021
Data	0x03	0xE8

7.2.15 Socket n TX Buffer Read Pointer Register (Sn_TX_RD) [0x0022-0x0023]

This register is used to obtain the transmitting and reading pointer of Socket n, which indicates the current position of the transmitting and reading pointer of CH394Q internal protocol stack. Users generally don't need to care about the value of this register, and CH394Q automatically manages the update.

After successful initialization of Socket n, this register points to the initial address of Socket n transmit buffer. After the user finishes writing the data into the SEND buffer and updates the pointer of Sn_TX_WR, CH394Q will automatically process the data transmission from Sn_TX_RD to Sn_TX_WR by transmitting the SEND command. See section 12.2.3 for the detailed operation.

7.2.16 Socket n TX Buffer Write Pointer Register (Sn_TX_WR) [0x0024-0x0025]

This register is used to get or set the transmitting and writing pointer of Socket n. It is necessary to get the transmitting and writing pointer of the current Socket n before each transmitting data process, and update the register value after writing the data to be transmitted. For details, please refer to Section 12.2.3 data transmitting process.

For example: The Socket n TX buffer write pointer is 2048.

Address	0x0024	0x0025
Data	0x08	0x00

7.2.17 Socket n RX Size Register (Sn_RX_RS) [0x0026-0x0027]

This register is used to obtain the size of data received by Socket n. Refer to Section 12.2.2 receiving data flow for details.

For example: Socket n receives data with a length of 1024.

Address	0x0026	0x0027
Data	0x04	0x00

7.2.18 Socket n RX Buffer Read Pointer Register (Sn_RX_RD) [0x0028-0x0029]

This register is used to get or set the Socket n receiving buffer reading pointer, indicating the current position of the user receiving the buffer reading pointer.

It is necessary to obtain the reading pointer of the current Socket n before receiving the data flow every time, and update the register value after reading the data. Refer to Section 12.2.2 receiving data flow for details.

For example: The read pointer of Socket n receiving buffer is 2048.

Address	0x0028	0x0029
Data	0x08	0x00

7.2.19 Socket n RX Buffer Write Pointer Register (Sn_RX_WR) [0x002A-0x002B]

This register is used to obtain the receiving buffer write pointer of Socket n, which indicates the current position of the receiving data write pointer of CH394Q internal protocol stack. Users generally don't need to care about the value of this register, and CH394Q automatically manages the update.

After successful initialization of Socket n, this register points to the initial address of Socket n receiving buffer. When CH394Q receives the required data from Ethernet and loads it into the receiving buffer, it will update the Sn_RX_WR pointer accordingly. Refer to Section 12.2.2 Receiving Data Flow for details.

7.2.20 Socket n Interrupt Enable Register (Sn_INTE) [0x002C]

This register is used to control which interrupt sources in Socket n can trigger interrupts. The default value is 0xFF. Each interrupt event enable bit corresponds to one bit of the Socket n interrupt register (Sn_INT).

When the interrupt event is generated, only when the corresponding bit in Sn_INTE is 1 and the nth bit of SINTE is 1 (Socket n interrupt is enabled) will CH394Q pull down the INT pin and generate an interrupt.

Table 7-13 Socket n interrupt enable register

Bit	Name	Description	Access	Default value
[7:5]	Reserved	Reserved	RO	-
4	INT_SEND_SUC	Transmit complete interrupt: Enabling this bit allows the generation of transmit complete interrupt.	RW	1
3	INT_TIMEOUT	Timeout interrupt: Enabling this bit allows a timeout interrupt to be generated.	RW	1
2	INT_RECV	Receive Interrupt: Enabling this bit allows a receive interrupt to be generated.	RW	1
1	INT_DISCONNECT	Disconnect Interrupt: Enabling this bit allows disconnection interrupt to be generated.	RW	1
0	INT_CONNECT	Connection Interrupt: Enabling this bit allows connection interruption.	RW	1

7.2.21 Socket n IP Fragment Register (Sn_IPF) [0x002D-0x002E]

This register is used to set the Socket n IP fragmentation field, and the default value is 0x4000.

For example: The Socket n IP fragmentation field is 0x4000 (No fragmentation).

Address	0x002D	0x002E
Data	0x40	0x00

7.2.22 Socket n Keep Alive Time Register (Sn_KEEPALIVE) [0x002F]

This register is used to set the Socket n Keep Alive time, which only takes effect in TCP mode. The default value is 0, indicating that the Keep Alive function is not activated. To enable auto Keep Alive, you need to set the value of this register to be greater than zero. The unit time of the register is 5 seconds, that is, when the register value is 1, the Keep Alive interval is 5 seconds, when it is 2, it is 10 seconds, and so on.

When Socket n is in TCP mode and has successfully established a connection with the peer (Sn_STA state is 0x17) after transmitting a packet and enters the idle (no data circulating) state, the system automatically sends KeepAlive packets according to the time interval specified by Sn_KEEPALIVE.

If the KeepAlive packet is continuously sent without acknowledgement from the other party, the system will trigger a timeout interrupt after the retransmission time limit is exceeded.

When Sn_KEEPALIVE is 0, users can manually transmit Keep Alive packets through the SEND_KEEP command. When Sn_KEEPALIVE is not 0, the SEND_KEEP command does not take effect.

For example: Socket n Keep Alive time registration is 4 (a Keep Alive packet is automatically sent every 20 seconds when it is idle).

Address	0x002F

Data	0x04

8. CH394Q IPv6 Function

This chapter mainly introduces the IPv6-related functions of CH394Q. If you do not use IPv6, you do not need to pay attention to the content of this chapter.

In the IPv6 mode, apart from the IPv4 protocol, CH394Q also supports ICMPv6 (including PING6, neighbor request NS and neighbor advertisement NA, router request RS, etc.), and supports data transmission in the IPv6 environment. CH394Q supports IPRAW4 and IPRAW6 transmission in IPv6 mode.

CH394Q can enter IPv6 mode by configuring the CHIP_CONFIG register (see section 7.1.17 for details). After switching to IPv6 mode, the overall operation mode is basically the same as the default IPv4 mode. On this basis, a number of IPv6 special registers and control bits are added to configure IPv6 addresses and related parameters, and support the transmitting and receiving operation of IPv6 packets. The normal operation and receiving and transmitting processes are still completed through the relevant registers in IPv4 mode, and the operation processes are basically the same.

In IPv6 mode, CH394Q divides the register space into multiple pages. The registers described in Chapter 7 are located on page 0. On the basis of the original register of page 0, some special registers are added in IPv6 mode, and IPv6-related control bits or related functions are added to some existing registers. In addition, the newly added pages 1 and 2 are dedicated to IPv6 register configuration. Users can switch register pages through the REG_PAGE register (this register is effective on any page). This chapter mainly introduces the newly added or changed IPv6-related registers in IPv6 mode, and the original registers already introduced in page 0 are not repeated here.

The general registers and Socket registers added or modified in IPv6 mode are used in the same way as in the conventional mode. Note that the register page number corresponds to the accessed register. Please ensure that the REG_PAGE is set correctly before use.

8.1 BASE Register

The BASE register can be used in any page without caring about the current page.

Table 8-1 BASE register

Offset address	Name	Access	Description	Reset value
0x003F	REG_PAGE	RW	Page selection register	0x00

8.1.1 Page Selection Register (REG_PAGE) [0x003F]

This register is used to configure the current register page, with the default value of 0 (page 0). When accessing other pages (such as page 1), you need to set this register to the corresponding page number (such as set to 1). The registers described in Chapters 7, 8.2 and 8.3 are all located on page 0, and the subsequent chapters (8.4 to 8.6) introduce the registers in page 1 and 2. Before using registers, make sure the current page is set correctly. This register is accessible in any page.

8.2 Page 0 Altered General-purpose Register

Table 8-2 Page 0 altered general-purpose register

Offset address	Name	Access	Description	Reset value

0x0015	GINT	RW	Global interrupt register	0x00

8.2.1 Global Interrupt Register (GINT) [0x0015]

This register is used to obtain the global interrupt state. When an interrupt event occurs, the corresponding bit of the GINT register will be set to 1. Writing 1 to this bit can clear the interrupt flag bit. If the enable bit of the interrupt in GINTE is also 1, the INT pin level will become low. Thereafter, once the corresponding bit of the event is cleared or the corresponding interrupt enable bit is set to 0 and SINT is 0, the INT pin returns to high level.

After enabling IPv6 mode, this register has added 3 new IPv6-related flag bits based on the original functions. The specific update is as follows:

Table 8-3 Global interrupt register (IPv6)

Bit	Name	Access	Access	Reset value
7	IP_CONFLI	IP conflict interrupt: This interrupt will occur when CH394Q detects that its own IP address is the same as other network devices within the same network segment.	RW1Z	0
6	UNREACH	Unreachable interrupt: When CH394Q receives the ICMP unreachable interrupt message, it saves the IP address, port, and protocol type of the unreachable IP packet into the unreachable information table and then generates this interrupt. When the microcontroller receives this interrupt, it can query the UNIP and UNPORT registers to get the unreachable information.	RW1Z	0
5	UNREACH_V6	Unreachable interrupt: When CH394Q receives the ICMPv6 unreachable interrupt message, saves the IP address, port, and protocol types of the unreachable IP packet to the unreachable information table, and then generates this interrupt. When the microcontroller receives this interrupt, it can query the UNIP6, UNPORT6 registers to get the unreachable information.	RW1Z	0
4	MP	Magic Packet Wakeup Interrupt: This interrupt is generated when the Magic Packet feature is enabled and a Magic Packet Wakeup packet is received over UDP.	RW1Z	0
3	RA	RA Receive Interrupt: This interrupt is generated when the CH394Q receives an RA message from the gateway. When the microcontroller receives this interrupt, it can query the VLT, PLT, PL, PF registers to get the	RW1Z	0

		prefix information.		
2	IP_CONFLI_V6	IPv6 conflict interrupt: When CH394Q has configured the local link address (LLA), it will automatically start the collision detection process. Within 1 second, if a neighbor advertisement (NA) reply to its own LLA address is detected, or a neighbor request (NS) with the same IP address sent by other devices is received, it means that the address has been used by other devices, and an address conflict is confirmed, thus triggering the interrupt.	RW1Z	0
[1:0]	Reserved	Reserved	RO	0

8.3 New and Changed Socket Registers in Page 0

Table 8-4 New and changed socket registers in page 0

Offset address	Name	Access	Description	Reset value
0x0000	Sn_MODE	RW	Socket n mode register	0x00
0x0001	Sn_CTRL	RW	Socket n control register	0x00
0x0003	Sn_STA	RO	Socket n status register	0x00
0x0014	Sn_IPPRO	RW	Socket n protocol register	0x00
0x0016	Sn_TTL	RW	Socket IP time-to-live register	0x80

8.3.1 Socket n Mode Register (Sn_MODE) [0x0000]

This register is used to configure the working mode of Socket n.

Table 8-5 Socket n mode register

Bit	Name	Access	Access	Reset value
7	MUL_MFEN	When Socket n is working in UDP4, UDP6, UDPD mode: 0: Disable multicast mode; 1: Enable multicast mode. When using multicast mode, you need to configure multicast IP and multicast MAC in advance through Sn_DIP or Sn_DIP6 and Sn_DMAC before opening the Socket. When Socket n is working in MACRAW mode: 0: Disable MAC address filtering; 1: Enable MAC address filtering. When using MACRAW mode, with MAC address filtering turned on, Socket n will only receive packets on the network whose destination MAC is the CH394Q's own MAC address as well as the broadcast address.	RW	0
6	BCASTB	Broadcast packet blocking bit:	RW	0

<table>
<tr><td></td><td></td><td>0: Do not block broadcast packets;
1: block broadcast packets.
After this bit 1, Socket n will not receive broadcast packets on the network in UDP4 and MACRAW modes.</td><td></td><td></td></tr>
<tr><td>5</td><td>NA_MV_MM</td><td>When Socket n is working in TCP4, TCP6, TCPD mode:
0: Disable no delay ACK;
1: Enable no delay ACK.
After enabling non-delay ACK, Socket n will reply to ACK as soon as possible without delay after receiving the data packet from the opposite end. Otherwise, it will wait for RTIME time before replying to ACK.
When Socket n is working in UDP4, UDPD multicast mode:
0: Use IGMP version 2;
1: Use IGMP version 1.
When Socket n is working in MACRAW mode:
0: Do not block multicast packets;
1: Block multicast packets.
After this position 1, in MACRAW mode, Socket n will not receive multicast packets on the network.</td><td>RW</td><td>0</td></tr>
<tr><td>4</td><td>UCASTB_MIP6B</td><td>When Socket n works in UDP4, UDP6 and UDP PD multicast modes:
0: Do not mask unicast packets;
1: Mask unicast packets.
When this bit is set, Socket n will not receive unicast packets on the network in UDP4, UDP6 and UDP multicast modes.
When Socket n is working in MACRAW mode:
0: Do not block IPv6 packets;
1: Block IPv6 packets.
After this position 1, in MACRAW mode, Socket n will not receive IPv6 packets on the network.</td><td>RW</td><td>0</td></tr>
<tr><td>[3:0]</td><td>PMD</td><td>Set Socket working mode:
<table>
<tr><th>[3:0]</th><th>Description</th></tr>
<tr><td>0000</td><td>Off</td></tr>
<tr><td>0001</td><td>TCP4</td></tr>
<tr><td>0010</td><td>UDP4</td></tr>
<tr><td>0011</td><td>IPRAW4</td></tr>
<tr><td>0100</td><td>MACRAW</td></tr>
<tr><td>1001</td><td>TCP6</td></tr>
<tr><td>1010</td><td>UDP6</td></tr>
<tr><td>1011</td><td>IPRAW6</td></tr>
</table></td><td>RW</td><td>0</td></tr>
</table>

		1101 TCPD 1110 UDPD *Note: Only Socket 0 can use MACRAW mode.*			

Note:

1. When the Socket is set to TCPD mode, IPv4 communication will be used when connecting to the server as a client using the CONNECT command, and IPv6 communication will be used when connecting to the server using the CONNECT6 command; when acting as a server, IPv4 communication will be performed if connected by an IPv4 client, and IPv6 communication will be performed if connected by an IPv6 client.

2. When the socket is set to UDPD mode, you can use the SEND or SEND6 command to send IPv4 or IPv6 packets respectively; after receiving the data, you can use the packet header information to determine whether its source is IPv4 or IPv6.

8.3.2 Socket n Control Register (Sn_CTRL) [0x0001]

This register is used to set the control command of Socket n. After the register is configured, it will be cleared automatically. After setting the control command, the control command usually needs to be executed for a certain period of time. Users can judge the command execution status through Sn_INT or Sn_STA registers.

Table 8-6 Socket n control register

<table>
<tr><th>Code</th><th>Name</th><th>Description</th></tr>
<tr><td>01H</td><td>OPEN</td><td>Open Socket n:
Initialize and open Socket n according to the mode set by Sn_MODE[3:0]. You can query the execution status of the command through Sn_STA. When the command is executed successfully, the corresponding Sn_STA value should be:
<table>
<tr><th>Sn_MODE[3:0]</th><th>Sn_STA</th></tr>
<tr><td>0000 (Off)</td><td>-</td></tr>
<tr><td>0001 (TCP4)</td><td>0x13 (INIT)</td></tr>
<tr><td>0010 (UDP4)</td><td>0x22 (UDP)</td></tr>
<tr><td>0011 (IPRAW4)</td><td>0x32 (IPRAW4)</td></tr>
<tr><td>0100 (MACRAW)</td><td>0x42 (MACRAW)</td></tr>
<tr><td>1001 (TCP6)</td><td>0x13 (INIT)</td></tr>
<tr><td>1010 (UDP6)</td><td>0x22 (UDP)</td></tr>
<tr><td>1011 (IPRAW6)</td><td>0x33 (IPRAW6)</td></tr>
<tr><td>1101 (TCPD)</td><td>0x13 (INIT)</td></tr>
<tr><td>1110 (UDPD)</td><td>0x22 (UDP)</td></tr>
</table>
</td></tr>
<tr><td>02H</td><td>LISTEN</td><td>Set Socket n as listen status:
This command takes effect when Socket n is in TCP4, TCP6, TCPD mode and in INIT state.
When Socket n is set to the LISTEN state, Socket n is a TCP Server waiting for the client to connect and the Sn_STA register becomes 0x14 (LISTEN). After the client connects successfully, the Sn_STA register value becomes 0x17 (ESTABLISHED) and Sn_INT[0] is set to 1.
If the connection fails Sn_STA register value becomes 0x00 (CLOSE) and Sn_INT[3] is set to 1.</td></tr>
</table>

04H	CONNECT	使 Enable Socket n to enter the connection mode and initiate a connection to the IPv4 server: This command is applicable when Socket n is in TCP4 or TCPD mode and the current state is INIT. After executing this command, Socket n will connect to the specified IPv4 server in TCP Client mode. When this command is executed, Socket n is connected to the server according to the values set by Sn_DIP and Sn_DPORT. When the connection is successful, the value of Sn_STA register becomes 0x17(ESTABLISHED) and Sn_INT[0] is set to 1. When the Socket connection fails: 1: ARP failure, unable to obtain the destination MAC address (Sn_INT[3] set to 1); 2: Connection timeout, after transmitting SYN packet and not receiving SYN/ACK packet from the other party, after retransmission timeout (Sn_INT[3] set to 1); 3: RST packet is received, and disconnection is initiated by the other party. The Sn_STA register value becomes 0x00 (CLOSE) when the above situations occur.
08H	DISCONNECT	Socket n Disconnect: This command takes effect when Socket n is in TCP4, TCP6, TCPD mode. Active shutdown: actively send FIN packets to the opposite device; Passive shutdown: when receiving FIN packets from the other side, execute this command to reply to the other side with a FIN packet. If the FIN packet does not receive an ACK response from the other party, Sn_INT[3] is set to 1 after the retransmission timeout, and the Sn_STA register value becomes 0x00 (CLOSE).
10H	CLOSE	Close Socket n: Directly closes the socket. When Socket n is in TCP4, TCP6, or TCPD mode, this command also directly closes the socket without transmitting a FIN packet.
20H	SEND	Socket n sends IPv4 data: Send out the data in the Socket n send buffer in TCP4, TCP6, TCPD, UDP4, UDPD, IPRAW4, and MACRAW modes, as detailed in Section 12.2.3 Data Sending Flow.
21H	SNED_MAC	Socket n sends data without ARP: This command takes effect only when Socket n is in UDP4 mode. The transmitting process is the same as that of the SEND command, but instead of obtaining the MAC address of the opposite end through ARP before transmitting the packet, the MAC address set by Sn_DMAC is used directly to send the packet.
22H	SEND_KEEP	Socket n sends KeepAlive packet: This command is effective only when Socket n is in TCP4, TCP6 and TCPD modes. Actively sends a heartbeat packet to the peer device once, and if no answer is

		obtained after the timeout, Sn_INT[3] is set to 1 and the Sn_STA register value changes to 0x00 (CLOSE).
40H	RECV	Socket n receives data: Complete the flow of Socket n receiving data, please refer to section 12.2.2 Data Receiving Flow for detailed steps.
84H	CONNECT6	Enable Socket n to enter the connection mode and initiate a connection to the IPv6 server: This command is applicable when Socket n is in TCP6 or TCPD mode and the current state is INIT. After executing this command, Socket n will connect to the specified IPv6 server in TCP Client mode. When this command is executed, Socket n is connected to the server according to the values set by Sn_DIP6 and Sn_DPORT. When the connection is successful, the value of Sn_STA register becomes 0x17(ESTABLISHED) and Sn_INT[0] is set to 1. When the socket connection fails: 1: ARP failure, unable to obtain the destination MAC address (Sn_INT[3] set to 1); 2: Connection timeout, send SYN packet and then can not receive the other party's SYN/ACK packet, after retransmission timeout (Sn_INT[3] set to 1); 3: RST packet is received, the other party is actively disconnected. The Sn_STA register value becomes 0x00 (CLOSE) when the above situations occur.
A0H	SEND6	Socket n sends IPv6 data: Send out the data in the Socket n send buffer in UDP6, UDPD, and IPRAW6 modes, as described in Section 12.2.3 Data Sending Flow.

8.3.3 Socket n Status Register (Sn_STA) [0x0003]

This register is used to obtain the current state of Socket n and the temporary state in operation.

Table 8-7 Socket n status register

Code	Name	Description
00H	CLOSE	Socket n is in the closed state.
13H	INIT	TCP Ready: When Socket n is in TCP4, TCP6, TCPD mode and OPEN is successful, Socket n is in the INIT state, only when Socket n is in the INIT state, the user can use the LISTEN or CONNECT command to make the next connection.
14H	LISTEN	TCP Listening: This state means that Socket n is already listening as a TCP server, waiting for a TCP client to connect. Sn_STA will change to 0x17 (ESTABLISHED) after successful connection.
17H	ESTABLISHED	Socket n establishes a connection: When Socket n is in the LISTEN state, as a TCP server mode, and is connected successfully by the TCP client of the opposite end; or is connected successfully as a TCP client using the CONNECT command, Sn_STA will change to 0x17 (ESTABLISHED).

		Sn_STA is 0x17 (ESTABLISHED) means the connection is successful, at this time you can transfer data normally.
1CH	CLOSE_WAIT	CLOSE_WAIT: When a Socket n TCP connection is established and a disconnect request packet (FIN packet) is received from the other side, Sn_STA will change to 0x10 (CLOSE_WAIT). At this time, the TCP connection is in a semi-disconnected state, to completely disconnect the connection you need to use the DISCONNT command, if you want to directly close, you can use the CLOSE command.
22H	UDP	Socket n is in UDP mode.
32H	IPRAW4	Socket n is in IPRAW4 mode.
33H	IPRAW6	Socket n is in IPRAW6 mode.
42H	MACRAW	Socket n is in MACRAW mode. *Note: Only Socket 0 can use MACRAW mode.*

8.3.4 Socket n IP Protocol Register (Sn_IPPRO) [0x0014]

This register is used to set the IP protocol number of Socket n (Protocol for IPv4 or Next Header field for IPv6). It is valid only in IPRAW4 and IPRAW6 modes and cannot be set to 0x06 (TCP) or 0x11 (UDP).

8.3.5 Socket n IP Time-to-Live Register (Sn_TTL) [0x0016]

This register is used to set the time-to-live field in the IP layer header: it corresponds to the TTL field in IPv4 and the Hop Limit field in IPv6. It should be set before Socket n performs an OPEN operation. The default value is 0x80, which does not need to be set by default.

8.4 General-purpose Registers in Page 1

Page 1 of CH394Q contains both general registers and Socket registers. This chapter only introduces the contents of general registers in page 1.

Table 8-8 General-purpose register in page 1

Offset address	Name	Access	Description	Reset value
0x0000 … 0x000F	LLA0 … LLA15	RW	Local link address register	0x00 … 0x00
0x0010 … 0x001F	GUA0 … GUA15	RW	Global unicast address register	0x00 … 0x00
0x0020 … 0x002F	GWIP60 … GWIP615	RW	IPv6 gateway register	0x00 … 0x00
0x0030 … 0x0033	RAVLT0 … RAVLT3	RO	RA valid lifetime register	0x00 … 0x00
0x0034 …	RAPLT0 …	RO	RA preferred time-to-live register	0x00 …

0x0037	RAPLT3			0x00
0x0038	RAPL	RO	RA prefix length register	0x00
0x0039	RAPF	RO	RA prefix flag register	0x00
0x003A	SUBPL	RW	Subnet prefix length register	0x00

8.4.1 Local Link Address Register (LLA) [0x0000-0x000F]

This register is used to set the local link address (LLA) of CH394Q, and the low byte address stores the high byte of LLA address.

For example, the LLA address is FE80::12:ABCD.

Address	0x0000	0x0001	0x0002	0x0003
Data	0xFE	0x80	0x00	0x00
Address	0x0004	0x0005	0x0006	0x0007
Data	0x00	0x00	0x00	0x00
Address	0x0008	0x0009	0x000A	0x000B
Data	0x00	0x00	0x00	0x00
Address	0x000C	0x000D	0x000E	0x000F
Data	0x00	0x12	0xAB	0xCD

8.4.2 Global Unicast Address Register (GUA) [0x0010-0x001F]

This register is used to set the global unicast address (GUA) of CH394Q, and the low byte address stores the high byte of GUA address.

For example, the GUA address is 2001:0db8::1234:5678.

Address	0x0010	0x0011	0x0012	0x0013
Data	0x20	0x01	0x0D	0XB8
Address	0x0014	0x0015	0x0016	0x0017
Data	0x00	0x00	0x00	0x00
Address	0x0018	0x0019	0x001A	0x001B
Data	0x00	0x00	0x00	0x00
Address	0x001C	0x001D	0x001E	0x001F
Data	0x12	0x34	0x56	0x78

8.4.3 IPv6 Gateway Register (GWIP6) [0x0020-0x002F]

This register is used to set the IPv6 gateway address of CH394Q, and the low byte address stores the high byte of the gateway address.

For example, the gateway address is FE80:: ABCD:1234.

Address	0x0020	0x0021	0x0022	0x0023
Data	0xFE	0x80	0x00	0x00
Address	0x0024	0x0025	0x0026	0x0027
Data	0x00	0x00	0x00	0x00
Address	0x0028	0x0029	0x002A	0x002B
Data	0x00	0x00	0x00	0x00
Address	0x002C	0x002D	0x002E	0x002F
Data	0xAB	0xCD	0x12	0x34

8.4.4 RA Valid Lifetime Register (RAVLT) [0x0030-0x0033]

This register is used to obtain the Valid Life time extracted from the prefix information of RA message, and the unit is seconds.

For example, the Valid Life time is 2592000 seconds (30 days).

Address	0x0030	0x0031	0x0032	0x0033
Data	0x00	0x27	0x8D	0x00

8.4.5 RA Preferred Lifetime Register (RAPLT) [0x0034-0x0037]

This register is used to obtain the Preferred Lifetime extracted from the prefix information of RA message, and the unit is seconds.

For example, the Preferred Lifetime is 604800 seconds (7 days).

Address	0x0034	0x0035	0x0036	0x0037
Data	0x00	0x09	0x3A	0x80

8.4.6 RA Prefix Length Register (RAPL) [0x0038]

This register is used to obtain the subnet prefix length extracted from the prefix information of the RA message.

8.4.7 RA Prefix Flag Register (RAPL) [0x0039]

This register is used to obtain the Prefix Flags extracted from the prefix information of the RA message.

8.4.8 Subnet Prefix Length Register (SUBPL) [0x003A]

This register is used to set the IPv6 subnet Prefix Length of the CH394Q, which is used to specify the number of bits of the network prefix. Usually, after receiving the RA interrupt, the user manually fills in the RAPL register value, or the user can configure it to other values by himself (not recommended).

8.5 Socket Registers in Page 1

Page 1 of CH394Q contains the Socket register, and its usage is the same as that of the Socket register in page 0. Make sure to set REG_PAGE to 1 before using it.

Table 8-9 Socket register in page 1

Offset address	Name	Access	Description	Reset value
0x0000 … 0x000F	Sn_DIP60 … Sn_DIP615	RW	Socket n destination IPv6 address register	0x00 … 0x00
0x0010	Sn_ESR	RO	Socket n extended status register	0x00

8.5.1 Socket n Destination IPv6 Address Register (Sn_DIP6) [0x0000-0x000F]

This register is used to get or set the destination IPv6 address of Socket n. There are 2 ways to use this register:

1: Get the destination IPv6 address, which is used in TCP6 Server, TCPD Server modes to store the IPv6 address of the TCP client after being successfully connected by the IPv6 peer.

2: Setting the destination IPv6 address, which is used to configure the destination IPv6 address in UDP6, UDPD, TCP6, TCPD client mode.

For example, the destination IPv6 address is FE80::AA:2233.

Address	0x0000	0x0001	0x0002	0x0003
Data	0xFE	0x80	0x00	0x00
Address	0x0004	0x0005	0x0006	0x0007
Data	0x00	0x00	0x00	0x00
Address	0x0008	0x0009	0x000A	0x000B
Data	0x00	0x00	0x00	0x00
Address	0x000C	0x000D	0x000E	0x000F
Data	0x00	0xAA	0x22	0x33

8.5.2 Socket n Extended Status Register (Sn_ESR) [0x0010]

This register is used to obtain the extended state of Socket n in TCP mode.

Table 8-10 Socket n extended status register

Bit	Name	Description	Access	Default value
[7:3]	Reserved	Reserved	RO	0
2	IP_OP	IP Protocol Version Selection In TCPD mode: 0: Use TCP4 mode; 1: Use TCP6 mode.	RO	0
1	TCP_OP	TCP mode indication: 0: TCP client; 1: TCP server.	RO	0
0	TCP6_SIP	Socket source address in TCP6 and TCPD modes: As a client, after the Socket connects to the IPv6 server successfully, this bit indicates the type of local address used: 0: Using the local link address; 1: Using the global unicast address. As a server, after being successfully connected by an IPv6 client, this bit indicates the type of server address used when the client connects: 0: Use local link address; 1: Use global unicast address.	RO	0

8.6 General-purpose Registers in Page 2

This chapter describes the contents of the general-purpose registers on page 2.

Table 8-11 General-purpose register in page 2

Offset address	Name	Access	Description	Reset value
0x0000 … 0x000F	PA0 … PA15	RO	RA prefix information register	0x00 … 0x00
0x0020 …	UNIP60 …	RO	IPv6 unreachable IP register	0x00 …

0x002F	UNIP615			0x00
0x0030 0x0031	UNPORT60 UNPORT61	RO	IPv6 unreachable port register	0x00 0x00
0x0032	ICMP6BK	RW	ICMP6 mask register	0x00

8.6.1 RA Prefix Information Register (PA) [0x0000-0x000F]

This register is used to obtain the prefix address information in the RA (Routing Advertisement) message sent by the router.

For example, the prefix address is 2001:0db8::.

Address	0x0000	0x0001	0x0002	0x0003
Data	0x20	0x01	0x0D	0xB8
Address	0x0004	0x0005	0x0006	0x0007
Data	0x00	0x00	0x00	0x00
Address	0x0008	0x0009	0x000A	0x000B
Data	0x00	0x00	0x00	0x00
Address	0x000C	0x000D	0x000E	0x000F
Data	0x00	0x00	0x00	0x00

8.6.2 RA Unreachable IP Register (UNIP6) [0x0020-0x002F]

This register is used to obtain the unreachable IPv6 address. When the CH394Q sends data to a port that is unreachable by an IPv6 device, the other party replies with an ICMPv6 packet (destination port unreachable). Upon receiving the unreachable packet, the CH394Q generates an IPv6 unreachable interrupt. The microcontroller can use this register to obtain the IPv6 unreachable address.

For example, the unreachable IP is FE80:: 1122:3344.

Address	0x0020	0x0021	0x0022	0x0023
Data	0xFE	0x80	0x00	0x00
Address	0x0024	0x0025	0x0026	0x0027
Data	0x00	0x00	0x00	0x00
Address	0x0028	0x0029	0x002A	0x002B
Data	0x00	0x00	0x00	0x00
Address	0x002C	0x002D	0x002E	0x002F
Data	0x11	0x22	0x33	0x44

8.6.3 IPv6 Unreachable Port Register (UNPORT6) [0x0030-0x0031]

This register is used to get the IPv6 unreachable port. When the CH394Q sends data to a port that is unreachable by an IPv6 device, the other party replies with an ICMPv6 packet (destination port unreachable). Upon receiving the unreachable packet, the CH394Q generates an IPv6 unreachable interrupt. The microcontroller can use this register to get the IPv6 unreachable port.

For example, the unreachable port is 2000.

8.6.4 IPv6 Mask Register (ICMP6BK) [0x0032]

This register sets the ICMPv6 packet masking.

Table 8-12 ICMP6 mask register

Bit	Name	Description	Access	Default value
[7:5]	Reserved	Reserved	RO	0
4	PING6	ICMPv6 PING request-response packet blocking bits: 0: Do not block request-response packets; 1: Block request-response packets.	RW	0
3	MLD	ICMPv6 MLD packet blocking bits: 0: Do not block MLD packets; 1: Block MLD packets.	RW	0
2	RA	ICMPv6 route announcement packet blocking bits: 0: Do not block route announcement packets; 1: Block route announcement packets.	RW	0
1	NA	ICMPv6 neighbor announcement packet blocking bit: 0: Do not block neighbor announcement packets; 1: Block neighbor announcement packets.	RW	0
0	NS	ICMPv6 neighbor request packet blocking bits: 0: Do not block neighbor request packets; 1: Block neighbor request packets.	RW	0

9. CH394Q Function Description

9.1 SPI Serial Interface

SPI synchronous serial interface signal line includes SPI chip selection input pin SCS, serial clock input pin SCK, serial data input pin SDI and serial data output pin SDO. Through SPI interface, CH394Q can be connected to the SPI serial bus of various MCUs, DSP and MCU with less wires, or make long-distance point-to-point connection.

SCS pin of CH394Q chip is driven by SPI chip selection output pin or common output pin of MCU, SCK pin is driven by SPI clock output pin SCK of MCU, SDI pin is driven by SPI data output pin MOSI of MCU, and SDO pin is connected to SPI data input pin MISO of MCU. For the hardware SPI interface, it is suggested that the SPI setting is CPOL=CPHA=0 or CPOL=CPHA=1, and the order of data bits is MSB first. The SPI interface of CH394Q also supports the MCU to communicate with the ordinary I/O pin analog SPI interface.

The SPI interface of CH394Q supports SPI mode 0 and SPI mode 3. CH394Q always inputs data from the rising edge of SPI clock SCK, and outputs data from the falling edge of SCK when output is allowed. The order of data bits is high, and 8 bits count as one byte.

Refer to Section 5 for SPI data frame format and examples, and Section 10.4.1 for SPI data frame timing requirements.

9.2 Other Hardware

The CH394Q chip internally integrates a 10M/100M Ethernet PHY, MAC, SPI-Slave controller, SRAM, high-speed MCU and PLL multiplier, and power supply power-on reset circuit.

The MDIRP, MDIRN, MDITP, and MDITN of CH394Q chip are the signal lines for Ethernet. PHY of CH394Q supports automatic conversion of MDI/MDIX lines.

The CH394Q chip has a built-in power-on reset circuit, and can also be controlled reset through the RSTB pin. The RSTB pin is used to input an asynchronous reset signal from the outside; when the RSTB pin is low, the CH394Q chip is reset; when the RSTB pin returns to high, the CH394Q will enter the initialization phase for about 15ms, during which the host prohibits operating CH394Q.

10. CH394Q Parameters

10.1 Absolute Maximum Ratings

(Critical or exceeding the absolute maximum value will probably cause the chip to work improperly or even be damaged)

Table 10-1 Absolute maximum value

Name	Parameter description	Min.	Max.	Unit
T_A	Ambient operating temperature	-40	85	℃
T_J	Junction temperature range	-40	100	℃
T_S	Ambient temperature during storage	-55	150	℃
AV_{DD33}	Operating power supply voltage	-0.4	4.0	V
V_{DDIO}	I/O supply voltage	-0.4	4.0	V
AV_{DDK}	Power decoupling end of core analog circuit	-0.4	1.5	V
DV_{DDK}	Power decoupling end of core digital circuit	-0.4	1.5	V
V_{ETH}	Voltage on ETH physical signal pin	-0.4	AV_{DD33}+0.4	V
V_{IN}	Input voltage on pin	-0.4	V_{DDIO}+0.4	V
$V_{ESD(HBM)}$	ESD electrostatic discharge voltage (HBM) for common I/O pins	6K		V
I_{IO}	Input current on I/O pin		20	mA
	Output current on I/O pin		20	

10.2 Electrical Parameters

Table 10-2 Electrical Parameter (AV_{DD33} = 3.3V, V_{DDIO} = 3.3V, T_A = 25℃)

Name	Parameter description		Min.	Typ.	Max.	Unit
AV_{DD33}	Operating power supply voltage		3.2	3.3	3.4	V
V_{DDIO}	I/O supply voltage		1.1	3.3	3.6	V
V_{IL}	I/O pin, input low voltage.		0		0.36* V_{DDIO}-0.21	V
V_{IH}	I/O pin, input high voltage.		0.40*VDDIO+0.49		V_{DDIO}	V
V_{OL}	Low level output voltage	V_{DDIO} = 3.3V 8mA sink current			0.4	V
		V_{DDIO} = 1.8V 4mA sink current			0.4	V
		V_{DDIO} = 1.2V 2mA sink current			0.3	V
V_{OH}	High level output voltage	V_{DDIO} = 3.3V 8mA output current	V_{DDIO}-0.4			V
		V_{DDIO} = 1.8V 4mA output current	V_{DDIO}-0.4			V
		V_{DDIO} = 1.2V 2mA output	V_{DDIO}-0.3			V

		current				
V_{hys}	Voltage hysteresis of I/O Schmitt trigger			150		mV
C_{IO}	I/O pin capacitance			5		pF
R_{PU}	Pull up equivalent resistance		30	40	55	kΩ
R_{PD}	Pull down equivalent resistance		30	40	55	kΩ
$t_{f(IO)out}$	Fall time of output high to low level	V_{DDIO} = 3.3V			6.5	ns
		V_{DDIO} = 1.8V			11	ns
		V_{DDIO} = 1.2V			20	ns
$t_{r(IO)out}$	Rise time of output low to high level	V_{DDIO} = 3.3V			6.5	ns
		V_{DDIO} = 1.8V			11	ns
		V_{DDIO} = 1.2V			20	ns

Note: All of the above are guaranteed by design parameters.

10.3 Power Consumption

Table 10-3 Power consumption (AV_{DD33} = 3.3V, V_{DDIO} = 3.3V, T_A = 25℃)

Symbol	Parameter	Condition (All current, with network regulator)	Typical	Unit
I_{DD0}	Supply current in transfer status	The link of 100BASE-TX path is successful and there are packets on the transceiver channel.	77.8	mA
		The link of 10BASE-TX path is successful and there are packets on the transceiver channel.	60.9	
I_{DD1}	Supply current in idle status	The link of 100BASE-TX path is successful and there are no data packets on the transceiver channel.	77.6	mA
		The link of 10BASE-TX path is successful and there are no data packets on the transceiver channel.	49.9	
I_{DD2}	Supply current in disconnect status	100BASE-TX and 10BASE-TX paths are not linked successfully and PHY is in auto-negotiation state.	55.9	mA
I_{DD3}	Supply current in PHY power-down status		15.4	

10.4 AC Electrical Characteristic and Timing

10.4.1 SPI Timing

Figure 10-1 SPI mode 0 timing figure

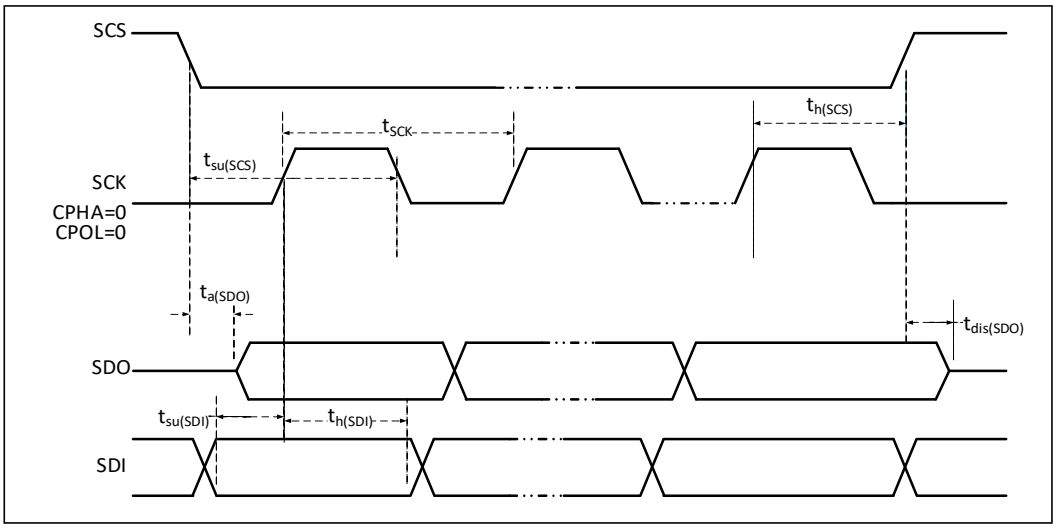


Figure 10-2 SPI mode 3 timing figure

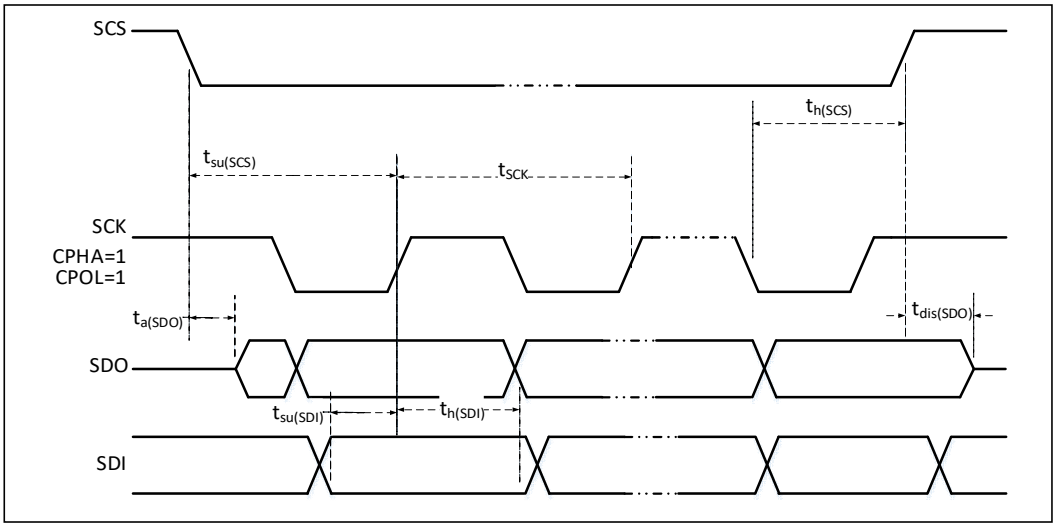


Figure 10-3 SPI serial interface figure

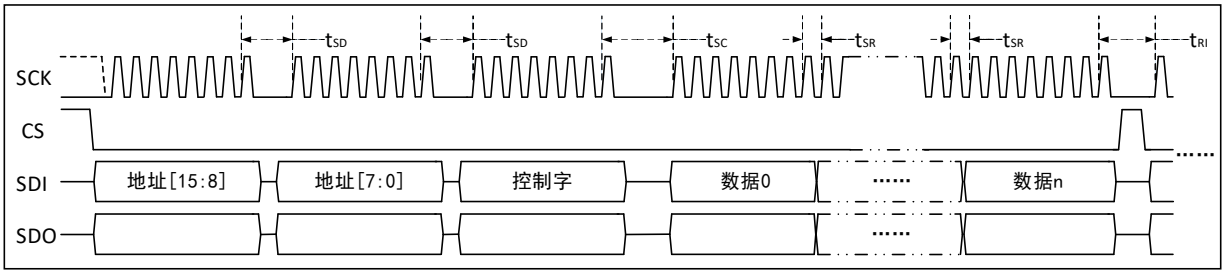


Table 10-4 SPI parameter table (AVDD33 = 3.3V, VDDIO = 3.3V, TA = 25℃)

Symbol	Parameter	Condition		Min.	Max.	Unit
f_{SCK}	SPI clock frequency(1)	V_{DDIO} = 3.3V			62	MHz
		V_{DDIO} = 1.8V			55	
		V_{DDIO} = 1.2V			30	
$t_{SU(SCS)}$	SCS setup time			17		ns
$t_{h(SCS)}$	SCS holding time			17		ns
$t_{SU(SDI)}$	Data input setup time			4		ns
$t_{h(SDI)}$	Data input holding time			2		ns
$t_{a(SDO)}$	Data output access time			0	8	ns
$t_{dis(SDO)}$	Data output disable time			0	10	ns
t_{SD}	Control word and offset address per byte interval			100		ns
t_{SC}	The required interval between the offset address and the first data byte	Write register		0		ns
		Read register		100		
		Write transceiver cache area		20		
		Read transceiver cache area		180		
t_{SR}	Required interval between data bytes	Write register		0		ns
		Read register		100		
		Read and write transceiver cache area	When f_{SCK} is not higher than 45M	0		
			When f_{SCK} is higher than 45M.	10		
t_{RI}	Interval required for data frame operation (2)	IPv4 mode (default) read & write register		0.3*N(3)-0.3		us
		IPv6 mode read & write register		N		
		Read and write transceiver cache area		0		

Note:

1. If the SPI rate is not higher than 5MHz, there is no need to consider additional delay parameters such as t_{SC}, t_{SD}, t_{SR}, etc., since the self-transmission delay of data transmission is sufficient, and the user doesn't need to add the delay manually.

2. When read/write operations are performed on registers, the host should wait for a tRI processing time after completing an operation, and the host is prohibited from operating other registers during the processing time.

3. N is the number of bytes in the data segment.

10.4.2 Oscillator and Crystal Oscillator Timing

Table 10-5 Parameters table of oscillator and crystal oscillator timing

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
TCKF	Crystal frequency	Recommended within 30ppm	24.999	25	25.001	MHz

TPWH	High clock pulse width		15	20	25	ns
TPWL	Low clock pulse width		15	20	25	ns

Note: The XI and XO pins have built-in two oscillation capacitors required by an external crystal with a load capacitor of 12pF, and only the crystal is needed externally; If an external crystal with a load capacitance of 20pF is selected, XI and XO need an additional oscillation capacitance of 15pF to the ground respectively.

10.4.3 Reset Timing

Table 10-6 Reset timing parameter table

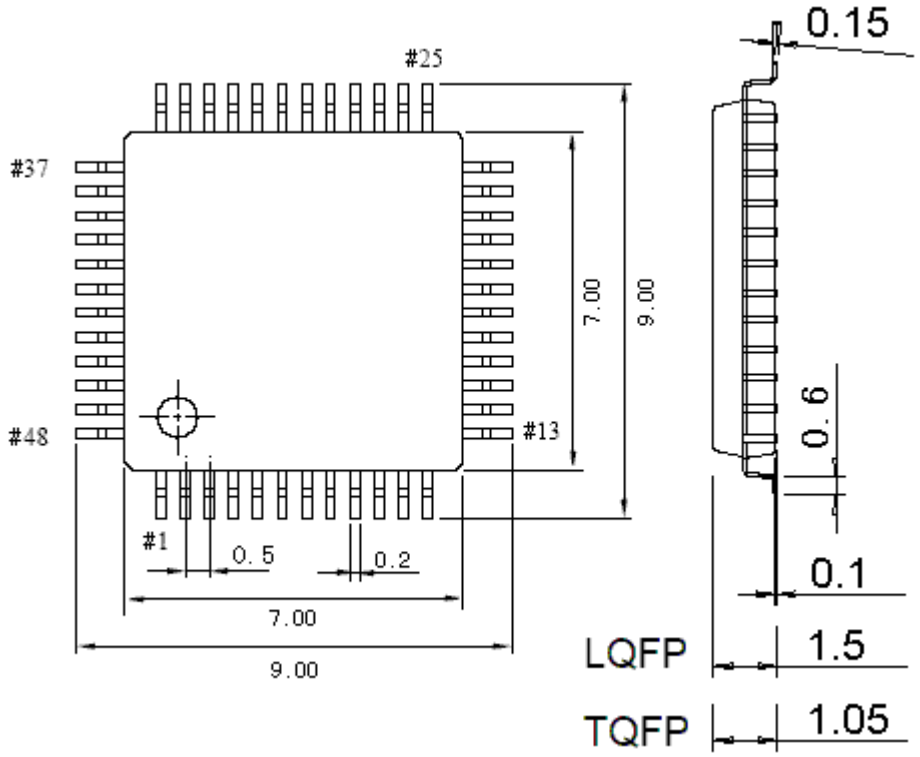
Symbol	Parameter	Min.	Typ.	Max.	Unit
$t_{RSTTEMPO}$	RSTB low level width	1			us
$t_{RSTTEMP1}$	RSTB high to host for operation	12	15	19	ms
$t_{RSTTEMP2}$	Reset command valid to host for operation	12	15	19	ms
$t_{RSTTEMP3}$	Power on and reset to the host for operation	27	30	35	ms

11. CH394Q Package Information

Note: All dimensions are in millimeters.

Pin center spacing is nominal with no error, except for dimensional error of no more than ±0.2mm.

11.1 LQFP48



12. CH394Q Application

12.1 Application Base

CH394Q integrates IPv4, IPv6, ARP, ICMP, ICMPv6, IGMP, UDP, TCP and other protocols.

TCP and UDP are two important transport layer protocols, both of which use IP as the network layer protocol.

TCP is a connection-oriented transmission, which can provide reliable byte stream transmission service.

UDP is a simple datagram-oriented transport layer protocol. Unlike TCP, UDP can't guarantee that the datagram can reach its destination accurately.

TCP provides highly reliable communication for network devices. Its work includes dividing the data handed to it by the application program into appropriate small pieces and handing it to the lower network layer, confirming the received packets, setting the timeout clock, etc. Because the transport layer provides highly reliable end-to-end communication, the application layer customers ignore all the details. UDP, on the other hand, provides a very simple service for the application layer, which is faster than TCP. It only transmits datagrams from one network terminal to another, but it does not guarantee that the datagrams can reach the other end. Any necessary reliability must be provided by the application layer.

IP is a protocol on the network layer, which is used by both TCP and UDP. Each set of data of TCP and UDP is transmitted in the network through the IP layer.

ICMP is an accessory protocol of IP protocol, which is used by IP layer to exchange error messages or other important information with other hosts or routers. For example, when CH394Q generates unreachable interrupt, it is through ICMP that error message switching is performed. PING also uses the ICMP protocol.

ICMPv6 is an important part of the IPv6 protocol and is responsible for delivering error messages and network diagnostic information. It supports key functions in the Neighbor Discovery Protocol (NDP), such as address resolution, neighbor reachability detection, PING, and so on.

IGMP is an Internet group management protocol, which is mainly used to multicast a UDP datagram to multiple hosts.

ARP is an address resolution protocol, which is used to convert the addresses used by IP layer and network interface layer.

12.2 IPv4 Application Reference Steps

This chapter introduces the operation process of CH394Q IPv4, and can refer to the routine for details.

12.2.1 Initialize CH394Q, necessary operations

Table 12-1 Initialize CH394Q operations

Step	Action	Operation register	Address	Operation
1	Set CH394Q MAC address	MAC	0x0009	Write MAC address
2	Set CH394Q IP address	IP	0x000F	Write IP address
3	Set gateway address	GWIP	0x0001	Write gateway address
4	Set the subnet mask address	SMIP	0x0005	Write subnet mask address

Step 1 is generally not required, the CH394Q is shipped with the IEEE assigned MAC address burned in.
Step 4 above is optional, the default subnet mask of 255.255.255.0 is generally not required.

12.2.2 Receive Data Process

When receiving data, we must:

Table 12-2 Receive data example

Step	Action	Operation register	Address	Operation
1	Get the length of the received data	Sn_RX_RS	0x0026	Read the receive length
2	Get the starting address of the data in the cache area	Sn_RX_RD	0x0028	Read the original address A_0
3	Take a specific length of data L0 from A0	-	A_0	Read data
4	Write the original address A0 + read length L0 back to Sn_RX_RD register	Sn_RX_RD	0x0028	Write A_0+L_0
5	Write RECV command to Sn_CTRL register	Sn_CTRL	0x0001	Write RECV command
6	Wait for command completion	Sn_CTRL	0x0001	Reading as 0x00 means execution is successful

12.2.3 Transmit Data Process

When transmitting data, we must:

Table 12-3 Transmit data example

Step	Action	Operation register	Address	Operation
1	Make sure there is enough space in the transmit buffer	Sn_TX_FS	0x0020	Read out the idle transmit buffer length L_0
2	Get the original address of the buffer where the data should be written	Sn_TX_WR	0x0024	Read source address A_0
3	Write a specific length of data, L1, to A0 (L1 needs to be less than L0)	-	A_0	Write data
4	Write the original address, A0 + the length of the write, L1, back to the Sn_TX_WR register	Sn_TX_WR	0x0024	Write A_0+L_1
5	Write the SEND command to the Sn_CTRL register	Sn_CTRL	0x0001	Write SEND command
6	Wait for the command to complete	Sn_CTRL	0x0001	Reading as 0x00 means execution is successful

12.2.4 Initialize Socket to UDP Mode

UDP is a simple, unreliable, data packet-oriented transportation layer protocol. It has a fast transmission speed and cannot guarantee that the data can reach its destination. The application layer must ensure the reliability and stability of the transmission.

After the microcontroller writes several bytes of data streams to CH394Q, CH394Q encapsulates the data stream

in the UDP data part for transmission. The maximum length that a UDP packet can send is 1472 bytes. If the length of the data stream written by the microcontroller is greater than 1472 bytes, CH394Q will automatically subpackage the data into a suitable size and send it.

When CH394Q receives the UDP message, it copies the UDP data to the Socket receiving buffer and generates an INT_RECV interrupt. After the microcontroller receives this interrupt, it can read the data in the buffer. Since CH394Q cannot provide flow control in UDP mode, it is recommended that the received data be read quickly and quickly to avoid data loss.

CH394Q supports UDP unicast, UDP broadcast and UDP multicast.

CH394Q will add an 8-byte information table to the head of the received data. The microcontroller can obtain the source information of the data packet based on the information table.

IP address	Port	Data package length	Data
4 Byte	2 Byte	2 Byte	N Byte

CH394Q directly sends data to the specified destination IP and port, and the microcontroller can set the destination IP and destination port before sending.
The steps to initialize Socket to UDP mode are as follows:

Table 12-4 Initialize socket to UDP mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket n to UDP mode	Sn_MODE	0x0000	Write 0x02
2	Set Socket n source port number	Sn_PORT	0x0004	Write source port number
3	Open Socket n	Sn_CTRL	0x0001	Write 0x01 command
4	Wait for the command to complete	Sn_CTRL	0x0001	Read as 0x00 means execution is successful

When Socket is set to UDP unicast or broadcast, before data transmission, just set Sn_DIP and Sn_DPORT as the corresponding unicast or broadcast addresses.

When the Socket is set to UDP multicast, Sn_DIP and Sn_DMAC should be set to the specified multicast IP address and MAC address respectively before the Socket is initialized, and the MUL_MFEN bit in the Sn_MODE register should be set to 1 to enable the multicast function. During the Socket connection, CH394Q will select the IGMP protocol version according to the NA_MV_MM bit in the Sn_MODE register, and send the group member JOIN message accordingly.

The user can CLOSE the Socket directly through the close command.

12.2.5 Initialize Socket to TCP Server Mode

The initialization steps are as follows:

Table 12-5 Initialize socket for TCP Server mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket n to TCP mode	Sn_MODE	0x0000	Write 0x01
2	Set Socket n source port number	Sn_PORT	0x0004	Write source port number
3	Open Socket n	Sn_CTRL	0x0001	Write 0x01 command

4	Wait for the command to complete	Sn_CTRL	0x0001	Read as 0x00 means successful execution
5	Set Socket n to listening mode	Sn_CTRL	0x0001	Write 0x02 command
6	Wait for the command to complete	Sn_CTRL	0x0001	Read as 0x00 means successful execution
7	Waiting for TCP client connection	-	-	-
8	TCP client connects successfully, and Socket n becomes ESTABLISHED	Sn_STA	0x0003	The value becomes 0x17

In TCP server mode, Socket will remain in LISTEN state, continuously listening for connection requests from the client without triggering the timeout interrupt mechanism until the SYN packet is received from the client. If the TCP handshake process is completed and a connection is established, CH394Q will trigger an INT_CONNECT interrupt; if the connection fails (i.e., the ACK packet acknowledged by the other party is not received), an INT_TIMEOUT interrupt will be triggered.

After successfully establishing the connection, the user can obtain the IP address and port number information of the client communicating with by querying the Sn_DIP and Sn_DPORT registers.

When CH394Q receives the FIN packet sent by the peer, it indicates that the other party requests to terminate the connection, and the system will generate an INT_DISCONNECT interrupt. Users can use the DISCONNECT instruction to send FIN packets, thereby completing the normal closing process of the connection.

Users can actively CLOSE the Socket through the DISCONNECT command or the close command.

12.2.6 Initialize Socket to TCP Client Mode

The initialization steps are as follows:

Table 12-6 Initialize socket for TCP Client mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket n to TCP mode	Sn_MODE	0x0000	Write 0x01
2	Set Socket n source port number	Sn_PORT	0x0004	Write source port number
3	Open Socket n	Sn_CTRL	0x0001	Write 0x01 command
4	Wait for the command to complete	Sn_CTRL	0x0001	Read as 0x00 means successful execution
5	Set destination IP	Sn_DIP	0x000C	Write destination IP
6	Set destination port	Sn_DPORT	0x0010	Write destination port
7	Start Socket connection	Sn_CTRL	0x0001	Write 0x04 command
8	Wait for the command to complete	Sn_CTRL	0x0001	Read as 0x00 means successful execution
9	Wait for the connection to succeed	-	-	-
10	TCP connection successful, socket n changes to ESTABLISHED state	Sn_STA	0x0003	The value becomes 0x17

After sending the connect command, CH394Q will actively send SYN packets to the server to initiate the TCP handshake process three times. If the TCP handshake process completes and a connection is established, the CH394Q will trigger the INT_CONNECT interrupt; if the connection fails (i.e., it fails to receive an acknowledged SYN/ACK packet from the other side), it will trigger the INT_TIMEOUT interrupt.

When CH394Q receives a FIN packet from the opposite end, it indicates that the other party requests to terminate the connection, and then the system will generate the INT_DISCONNECT interrupt. The user can use the DISCONNECT instruction to send the FIN packet to complete the normal shutdown process of the connection.

The user can actively close the socket by using the DISCONNECT command or the CLOSE command.

12.2.7 Initialize Socket to MACRAW Mode

The initialization steps are as follows:

Table 12-7 Initialize socket for MACRAW mode steps

Step	Action	Operation register	Address	Operation
1	Set Socket0 to MACRAW mode	Sn_MODE	0x0000	Write 0x04
2	Open Socket0	Sn_CTRL	0x0001	Write 0x01 command
3	Wait for the command to complete	Sn_CTRL	0x0001	Read as 0x00 means successful execution

Only Socket 0 supports MACRAW mode. In MACRAW mode, the CH394Q transparently transmits data between the Ethernet and the microcontroller, and does not TCP/IP encapsulate the data. in MACRAW mode, Socket 0 will not receive packets accepted by other sockets.

The CH394Q will add a 2-byte information table to the header of the received data.

Data package length	Data
2 Byte	N Byte

12.3 IPv6 Application Reference Steps

This chapter introduces the operation process of CH394QIPv6 mode, please refer to the routine for details.

12.3.1 Initialize CH394Q IPv6

In IPv6 mode, CH394Q can configure both IPv4 and IPv6 addresses. Please refer to Section 12.2.1 for the configuration method of IPv4 address. This article only introduces the configuration of IPv6 address.

Prefix information for IPv6 addresses is usually assigned by RA (Router Advertisement) messages. When the CH394Q receives an RA message, the Global Interrupt Register (GINT) generates an RA interrupt. At this time, users can obtain subnet prefix related information through the Subnet Prefix Length Register (RAPL), RA Prefix Flag Register (RAPF), and RA Prefix Information Register (PA), etc., refer to Section 12.3.1 for details. The following table lists the specific methods for configuring IPv6 addresses.

Table 12-8 Initialize CH394Q steps

Step	Action	Operation register	Address	Operation
1	Set the local link address of the CH394Q	LLA	Page 1 0x0000	Write LLA
2	Set the global unicast address of the CH394Q	GUA	Page 1 0x0010	Write GUA
3	Set the subnet prefix length of the CH394Q	SUBPL	Page 1 0x003A	Write subnet prefix length

4	Set the IPv6 gateway address	GWIP6	Page 1 0x0020	Write gateway address

12.3.2 CH394Q IPv6 Neighbors Found out

CH394Q will automatically send RS (Router Solicitation) packets after entering IPv6 mode, retransmitting RCN (Retransmission Count Register) times with RTIME (Retransmission Time Register) time interval. If a router in the LAN responds to the correct RA (Router Advertisement) packet, CH394Q will parse the RA packet data, store the parsed information (prefix address, length, survival time, etc.) into PA, RAPL, RAPF, RAVLT, etc. registers, and report the RA reception interrupt in the Global Interrupt Register (GINT). At this time, the user can set the global unicast address of CH394Q according to the prefix information. Users can also configure the address by themselves without using RA information.

When the CH394Q finishes configuring the local link address (LLA), it automatically starts the conflict detection process (DAD). Within 1 second, if it detects a Neighbor Announcement (NA) reply to its own LLA address, or receives a Neighbor Request (NS) sent by another device using the same IP address that means the address has been used by another device, that is, it confirms that an address conflict has occurred and reports an address conflict interrupt in the Global Interrupt Register (GINT). socket sending and receiving operations are invalidated during the DAD process (1 second).

12.3.3 IPv6 Receive Data Process

The socket can start receiving data only after the DAD (Duplicate Address Detection) process is completed and there are no address conflicts. see Section 12.3.2 for details of the DAD process.

The receive process is identical to IPv4 mode, see Section 12.2.2 for details.

12.3.4 IPv6 Transmit Data Process

The socket can start sending data only after the DAD (Duplicate Address Detection) process is completed and there are no address conflicts. see Section 12.3.2 for details of the DAD process.

The sending process is exactly the same as in IPv4 mode, see Section 12.2.3 for details.

12.3.5 Initialize Socket to UDP Mode

In IPv6 mode, the CH394Q supports configuring the socket to UDP4, UDP6 and UDPD modes.

The UDP creation process is basically the same as in IPv4 mode, which can be referred to Section 12.2.4. There are differences only in some detailed operations, which are described below:

Table 12-9 IPv6 UDP mode introduction

Mode	Description
UDP4	1、 The usage is exactly the same as that in IPv4 mode. Socket creation, data sending and receiving, and multicast operation are the same. 2、 Can receive packets sent to the local IPv4 address, receive data format refer to Figure 12-1.
UDP6	1、 The creation process is basically the same as the UDP4 mode, only need to set the Socket Mode Register (Sn_MODE) to 0x0A (UDP6); 2、 Before sending the packet, you need to configure the destination IPv6

	address register (Sn_DIP6), use the SEND6 (0xA0) command to send; 3、 Only receive packets addressed to the IPv6 address of the local machine, the format of the packet received refer to Figure 12-1.
UDPD	1、 The creation process is basically the same as the UDP4 mode, only need to set the Socket Mode Register (Sn_MODE) to 0x0E (UDPD); 2、 If you send data to the IPv4 address, you need to configure Sn_DIP, use the SEND (0x20) command; 3、 If you send data to the IPv6 address, you need to configure Sn_DIP6, use the SEND6 (0xA0) command; 4, this mode can receive packets sent to both local IPv4 and IPv6 addresses at the same time. 0xA0) command; 4、 This mode can receive packets sent to the local IPv4 and IPv6 addresses at the same time, refer to Figure 12-1 for the reception format.

CH394Q will add an 8 or 20Byte information table to the header of the received data, and the user can obtain the source information of the data packet based on the information table.

Figure 12-1 Receive packet format in UDP mode

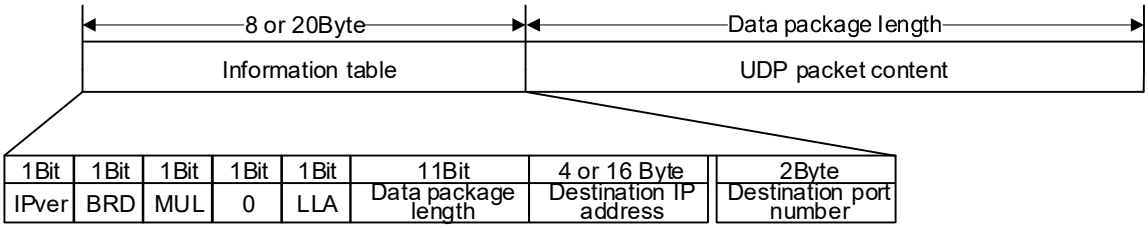


Table 12-10 UDP data package parameters

Name	Description	Length
IPver	0: Data source is IPv4 packet 1: Data source is IPv6 packet	1Bit
BRD	0: Data source is a non-broadcast packet 1: Data source is a broadcast/node-wide multicast packet	1Bit
MUL	0: Data source is a non-multicast packet 1: Data source is a multicast packet	1Bit
0	Reserved, always be 0	1Bit
LLA	0: From global unicast address 1: From local link address	1Bit
Data package length	Receive data package length	11Bit
Destination IP address	Destination IPv4 address when UDP4 packet is received (4 bytes);	4Byte or

	Destination IPv6 address when UDP6 packet is received (16 bytes)	16Byte
Destination port number	Destination port number	2Byte

The multicast and broadcast configuration method of CH394Q in IPv6 mode is the same as IPv4 mode. When using regular or multicast, only the destination IPv4 or IPv6 register and the destination MAC register should be set to the corresponding addresses.

When using or multicast, the destination IP and MAC addresses should be set in advance.

12.3.6 Initialize Socket to TCP Server Mode

In IPv6 mode, the CH394Q supports configuring the Socket to TCP4, TCP6 and TCPD modes.

The creation process of TCP Server is basically the same as in IPv4 mode, which can be referred to Section 12.2.5. There are differences only in some detailed operations, which are described below:

Table 12-11 IPv6 TCP Server introduction

Mode	Description
TCP4	1、 The usage is exactly the same as that in IPv4 mode. 2、 Socket creation, data sending and receiving, listening and other operations are exactly the same as in IPv4 mode; 3、 It can be accessed by the client connecting to the local IPv4 address; 4、 The data sending and receiving process is exactly the same as in IPv4 mode.
TCP6	1、 The usage is exactly the same as that in IPv4 mode. Only need to set the Socket Mode Register (Sn_MODE) to 0x09 (TCP6). 2、 Socket creation, data sending and receiving, listening and other operations are exactly the same as in IPv4 mode; 3、 Can be connected to the local IPv6 address of the client access; 4、 The user can be Socket n Extended Status Register (Sn_ESR) bit 0 to determine whether the client is connected to the local link address or a global unicast address; 5、 Obtain the opposite end of the IP need to be through the Sn_DIP6 register; 6、 The data sending and receiving process is exactly the same as IPv4 mode.
TCPD	1、 The usage is exactly the same as that in IPv4 mode. Only need to set Sn_MODE to 0x0D (TCPD); 2、 Socket creation, data sending and receiving, listening and other operations are exactly the same as in IPv4 mode; 3、 Support for IPv4 or IPv6 clients to connect to the local machine; 4、 The user can determine whether the client is using an IPv4 connection or an IPv6 connection through the Sn_ESR registers, and distinguish whether the IPv6 connection is a local link address or a global unicast address;

	5、If Sn_ESR finds out that it is an IPv4 connection, it can obtain the IP of the opposite end through Sn_DIP register; if it is an IPv6 connection, it can obtain the IP of the opposite end through Sn_DI6P register; 6、The process of data sending and receiving is exactly the same as that of the IPv4 mode.

12.3.7 Initialize Socket to TCP Client Mode

In IPv6 mode, the CH394Q supports configuring the Socket to TCP4, TCP6 and TCPD modes.

The creation process of TCP Client is basically the same as in IPv4 mode, which can be referred to Section 12.2.6. There are differences only in some detailed operations, which are described below:

Table 12-12 IPv6 TCP Client introduction

Mode	Description
TCP4	1、The usage is exactly the same as that in IPv4 mode. 2、Socket creation, data sending and receiving, listening and other operations are exactly the same as in IPv4 mode; 3、It can be accessed by the client connecting to the local IPv4 address; 4、The data sending and receiving process is exactly the same as in IPv4 mode.
TCP6	1、The usage is exactly the same as that in IPv4 mode. Only need to set the Socket Mode Register (Sn_MODE) to 0x09 (TCP6). 2、Socket creation, data sending and receiving, listening and other operations are exactly the same as in IPv4 mode; 3、Can be connected to the local IPv6 address of the client access; 4、Send the CONNECT6 (0x84) command to initiate the connection to the peer IPv6 server; 5、The data sending and receiving process is exactly the same as IPv4 mode.
TCPD	1、The usage is exactly the same as that in IPv4 mode. Only need to set Sn_MODE to 0x0D (TCPD); 2、Socket creation, data sending and receiving, listening and other operations are exactly the same as in IPv4 mode; 3、Set the destination IPv4 address via Sn_DIP, or set the destination IPv6 address via Sn_DIP6; 4、Use the CONNECT6 (0x84) command to connect to an IPv6 server, or you can connect to an IPv4 server using the CONNECT (0x04) command; 5、The process of data sending and receiving is exactly the same as that of the IPv4 mode.

12.3.8 Initialize Socket to MACRAW Mode

Exactly the same as in IPv4 mode, see Section 12.2.7 for details.

12.3.9 Initialize Socket to IPRAW Mode

The initialization steps are as follows:

Table 12-13 Initialize socket for IPRAW mode steps

Step	Action		Operation register	Address	Operation
1	Set Socket n to	IPRAW4 mode	Sn_MODE	0x0000	Write 0x03
		IPRAW6 mode	Sn_MODE	0x0000	Write 0x0B
2	Set Socket n IP protocol type		Sn_IPPRO	0x0014	Set the protocol type
3	Open Socket n		Sn_CTRL	0x0001	Write 0x01 command
4	Wait for the command to complete		Sn_CTRL	0x0001	Read as 0x00 means execution is successful

The IPRAW protocol type does not cover the TCP and UDP protocols.

In IPRAW4 mode:

In this mode Socket n can directly send and receive IPv4 packets. When the protocol type of Socket n is configured as ICMP (identifier 0x01), Socket n will capture all ICMP datagrams received by the CH394Q, in which case the CH394Q will not automatically respond to ICMP datagrams from the opposite end (e.g., PING requests). All received ICMP datagrams will be stored in Socket n's receive buffer.

The CH394Q will add a 6-byte information table to the header of the received data, which the microcontroller can use to obtain information about the source of the packet.

IP address	Data package length	Data
4 Byte	2 Byte	N Byte

In IPRAW6 mode:

In this mode Socket n can directly send and receive IPv6 packets. When the protocol type of Socket n is configured as ICMPv6 (Next Header is 0x58), Socket n will capture all ICMPv6 packets received by CH394Q that are not automatically processed by the protocol stack. Packet types that are not expected to be processed by the protocol stack can be masked via the ICMP6 Mask Register (ICMP6BK). When a packet of this type is received after masking, the packet is deposited into the receive buffer of Socket n. Other ICMPv6 packets not marked by the ICMP6BK register are also deposited into the receive buffer of Socket n. The ICMPv6 packet is not marked by the ICMP6BK register.

The CH394Q will add an 18-byte information table to the header of the received data, which allows the user to get the source information of the packet.

Figure 12-2 Receive data packet format in IPRAW6 mode

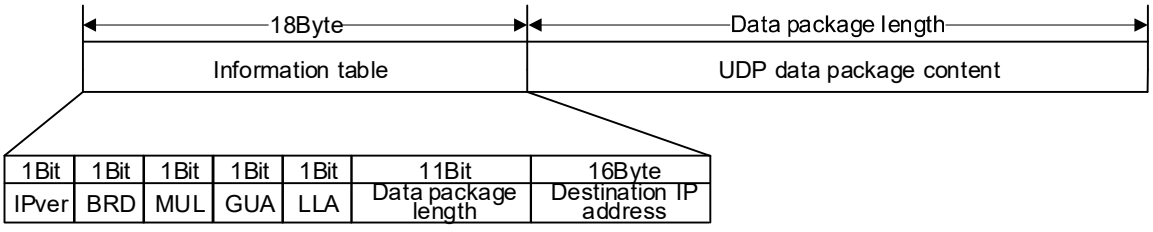
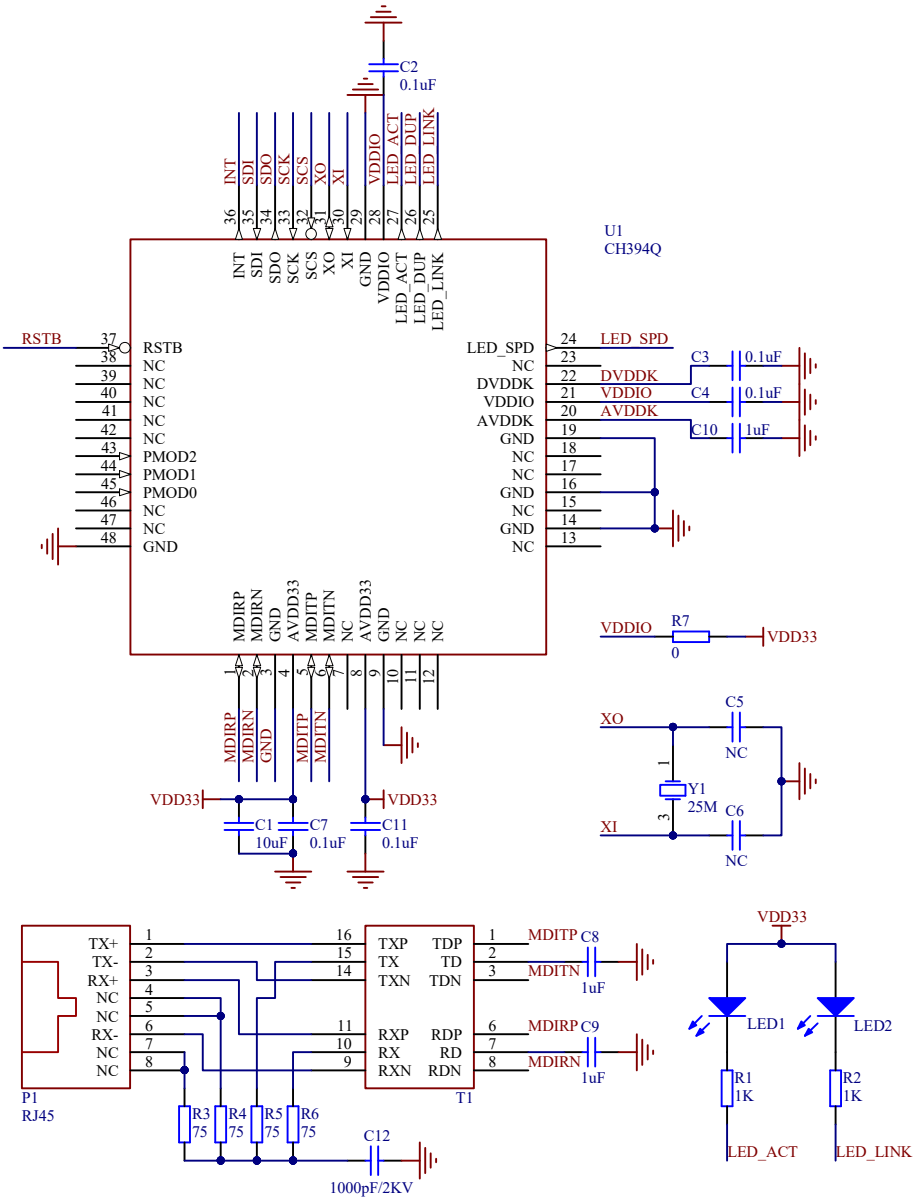


Table 12-14 IPRAW6 data package parameters

Name	Description	Length
IPver	Fixed to 1, the data source is IPv6 packet	1Bit
BRD	0: Data source is a non-broadcast packet 1: Data source is a broadcast/node-wide multicast packet	1Bit
MUL	0: Data source is a non-multicast packet 1: Data source is a multicast packet	1Bit
GUA	0: Other 1: From global unicast address	1Bit
LLA	0: Other 1: From local link address	1Bit
Data packet length	Received packet length	11Bit
Destination IP address	Destination IPv6 address	16Byte

12.4 CH394Q Hardware Circuit Design

Figure 12-3 Hardware reference circuit



CH394Q has built-in partial oscillation capacitance of crystal Y1, and C5 and C6 can be adjusted according to crystal parameters. For Y1 with a load capacitance of 12pF, C5 and C6 are not needed; For Y1, C5 and C6 with a load capacitance of 20pF, 15pF each is recommended.

CH394Q has built-in Ethernet 50Ω impedance matching resistor, so don't connect 49.9Ω or 50Ω resistor externally, which is equivalent to voltage driving.

T1 is an ethernet network transformer, and its center tap is grounded through capacitors C8 and C9 respectively, so don't connect any power supply.